

Speaker Recognition  
Office Automation

Deep Learning Project

Final Report

Contents

[Model Development 2](#_Toc33736595)

[Using **Cosine Distance** Calculation 3](#_Toc33736596)

[Momentum 3](#_Toc33736597)

[Time-based Decay Rate 3](#_Toc33736598)

[Optimizer 3](#_Toc33736599)

[Using **Weighted L1 Distance** Calculation 3](#_Toc33736600)

[Optimizer and Learning Rate 3](#_Toc33736601)

[Loss Function 4](#_Toc33736602)

[Using **Euclidean Distance** Calculation 4](#_Toc33736603)

[CNN model with more layers 4](#_Toc33736604)

[Training accuracy and loss 4](#_Toc33736605)

[CNN model with reduced hidden layers and with Euclidean distance 5](#_Toc33736606)

[Optimizer and Learning Rate 5](#_Toc33736607)

[Loss Function 5](#_Toc33736608)

[Using Euclidean Distance with **Contrastive Loss** 6](#_Toc33736609)

[Using Euclidean Distance with **Triplet Loss** 6](#_Toc33736610)

[Comparison of model performances 6](#_Toc33736611)

[Model Deployment 7](#_Toc33736612)

[API via two routes 8](#_Toc33736613)

[References 8](#_Toc33736614)

Final Report

# Model Development

The basis for our model is One Shot Learning using Siamese Network. This network essentially learns a similarity function as opposed to traditional classification. In large organizations scenario, where there is higher frequency of joiners and leavers, this method eliminates the need to collect numerous samples of new voices to retrain the network – just one voice file is needed.

MFCC features (max 40) are extracted for all the voice files from the training and testing datasets. Samples were created by randomly selecting the subjects and pairing the MFCC feature records of each voice file of the subject with own and other subjects MFCC records. For the pairs that contain MFCC records of the same subject the target is set to 1 and for unrelated pairs, the targets are set to 0. The pairs and targets are then used for training the model.

Due to the sheer amount of processing and hardware limitations, we have adopted Random Pairs comparison where a random set of 3 subjects at most – were compared against, at each iteration. We ran into ResourceExhaustedError when running beyond the maximum.

We have constructed our Neural Networks model, testing with five, as well as three hidden layers. Ultimately, models with three hidden layers performed better.

We have used the following model definition for our training:

|  |
| --- |
|  |

On first running each of our models, we hit upon a similar issue. Accuracy was stuck at the same value either from the start or at some point. To circumvent being stuck at local minima, we introduced BatchNormalization. This effectively improved some of our models.

## Using **Cosine Distance** Calculation

BatchNormalization has no effect on Cosine Distance. Accuracy was stuck at 0.3333 while Validation accuracy was at 0.5000, while losses barely moved at a high 11.700-11.800. Several attempts were made throughout fine-tuning including adding Conv2D layers, alternative Activation functions and Optimizer hyperparameter values. Loss was kept at Binary CrossEntropy throughout. The following are key steps that effect some notable change:

### Momentum

Momentum is a popular method to kick a model out of local minima. Switched the optimizer to Stochastic Gradience Descent, specifying momentum.  
Result: No changes to accuracies but loss kicked off at 0.7585 and slowly but gradually decreases at a rate or 0.0001

|  |
| --- |
| momentum = 0.8  optimizer = SGD(lr=learning\_rate, momentum=momentum, nesterov=False) |

### Time-based Decay Rate

Result: Accuracy rises to 0.5000. Validation Accuracy remains at 0.5000, Loss continues decreasing at a slow rate.

|  |
| --- |
| learning\_rate = 0.1  epochs = 20  decay\_rate = learning\_rate / epochs  optimizer = SGD(lr=learning\_rate, momentum=momentum, decay=decay\_rate, nesterov=False) |

### Optimizer

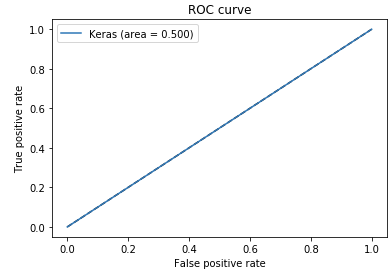


Figure 1 ROC for Cosine Distance

Optimizer was switched back to Adam since Adam is more aggressive at finding minimal loss. Thankfully, decay is still supported for Adam optimizer in TF2.  
Result: Accuracy and Validation accuracy unchanged. Loss now decreases at an average rate of 0.01.

|  |
| --- |
| optimizer = Adam(lr=learning\_rate, decay=decay\_rate,  amsgrad=False) |

## Using **Weighted L1 Distance** Calculation

### Optimizer and Learning Rate

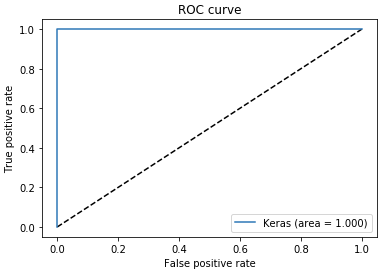


Figure 2 ROC for Weighted L1 Distance

Optimizers are used in neural networks to reduce the loss.

* The model was compiled using the Adam optimizer /SGD . We found that Adam optimizer performs better for our model than and SGD.

Learning rate controls how much we are adjusting the weights of our network with respect the loss gradient.

* We have kept the Learning rate was low because with high learning rate, the model took very long time to converge. as it was found that with high learning rate, the model took a lot of time to converge.

|  |
| --- |
| optimizer = Adam(lr = 0.00006)  model.compile(loss="binary\_crossentropy", metrics=['accuracy'], optimizer=optimizer) |

### Loss Function

* We used loss function to compute the gradients, using which we are going to update the weight and bias of Siamese Network. We have used Binary CrossEntropy, contrastive loss and Triplet loss functions in our training.
* We have found that by using Binary CrossEntropy our model can train with high accuracy.

|  |
| --- |
| model.compile(loss="binary\_crossentropy", metrics=['accuracy'], optimizer=optimizer) |

## Using **Euclidean Distance** Calculation

### CNN model with more layers

* We tried to train the NN model with many layers, to check if our accuracy improves performs better when there are more hidden layers, as it is expected that more hidden layers may increase the accuracy.
* We wanted to avoid the Dying Relu problem and used LeakyRelu, as activation function and Euclidean distance measure for distance calculation. It was giving out of memory error with VM and Google Colab
* We changed the activation to Relu, but with same number of hidden layers like below. It didn’t give ‘Out of memory’ error with Google Colab and we trained our model in Google Colab.
* We used Adam as optimizer and with lower learning rate.

|  |
| --- |
|  |

### Training accuracy and loss

* During the initial stages of training, training accuracy was 0.70 but the validation accuracy was very low. After few epochs it started moving closer to training accuracy.
* After a few epochs, the training accuracy showed very little raise and the loss was reducing at regular pace.
* Once validation accuracy reached 0.70, raise in the accuracy was almost negligible or nil.
* The time used to train the model was very long.

|  |
| --- |
|  |

The below image shows the range of training and validation accuracy after the validation accuracy reaches the range of 0.70

|  |
| --- |
|  |

Figure Preliminary Accuracy and Loss for Euclidian

As we are not satisfied with the results, we reduced some hidden layers and retrained our model again

### CNN model with reduced hidden layers and with Euclidean distance

![A screenshot of a cell phone

Description automatically generated](data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEAkACQAAD/4REQRXhpZgAATU0AKgAAAAgABAE7AAIAAAAXAAAISodpAAQAAAABAAAIYpydAAEAAAAuAAAQ2uocAAcAAAgMAAAAPgAAAAAc6gAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAFBhZG1hcHJpeWEsIFNyaW5pdmFzYW4AAAAFkAMAAgAAABQAABCwkAQAAgAAABQAABDEkpEAAgAAAAMxMQAAkpIAAgAAAAMxMQAA6hwABwAACAwAAAikAAAAABzqAAAACAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAMjAyMDowMjoyNyAyMDoyODo1MgAyMDIwOjAyOjI3IDIwOjI4OjUyAAAAUABhAGQAbQBhAHAAcgBpAHkAYQAsACAAUwByAGkAbgBpAHYAYQBzAGEAbgAAAP/hCylodHRwOi8vbnMuYWRvYmUuY29tL3hhcC8xLjAvADw/eHBhY2tldCBiZWdpbj0n77u/JyBpZD0nVzVNME1wQ2VoaUh6cmVTek5UY3prYzlkJz8+DQo8eDp4bXBtZXRhIHhtbG5zOng9ImFkb2JlOm5zOm1ldGEvIj48cmRmOlJERiB4bWxuczpyZGY9Imh0dHA6Ly93d3cudzMub3JnLzE5OTkvMDIvMjItcmRmLXN5bnRheC1ucyMiPjxyZGY6RGVzY3JpcHRpb24gcmRmOmFib3V0PSJ1dWlkOmZhZjViZGQ1LWJhM2QtMTFkYS1hZDMxLWQzM2Q3NTE4MmYxYiIgeG1sbnM6ZGM9Imh0dHA6Ly9wdXJsLm9yZy9kYy9lbGVtZW50cy8xLjEvIi8+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczp4bXA9Imh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC8iPjx4bXA6Q3JlYXRlRGF0ZT4yMDIwLTAyLTI3VDIwOjI4OjUyLjExMzwveG1wOkNyZWF0ZURhdGU+PC9yZGY6RGVzY3JpcHRpb24+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczpkYz0iaHR0cDovL3B1cmwub3JnL2RjL2VsZW1lbnRzLzEuMS8iPjxkYzpjcmVhdG9yPjxyZGY6U2VxIHhtbG5zOnJkZj0iaHR0cDovL3d3dy53My5vcmcvMTk5OS8wMi8yMi1yZGYtc3ludGF4LW5zIyI+PHJkZjpsaT5QYWRtYXByaXlhLCBTcmluaXZhc2FuPC9yZGY6bGk+PC9yZGY6U2VxPg0KCQkJPC9kYzpjcmVhdG9yPjwvcmRmOkRlc2NyaXB0aW9uPjwvcmRmOlJERj48L3g6eG1wbWV0YT4NCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgPD94cGFja2V0IGVuZD0ndyc/Pv/bAEMABwUFBgUEBwYFBggHBwgKEQsKCQkKFQ8QDBEYFRoZGBUYFxseJyEbHSUdFxgiLiIlKCkrLCsaIC8zLyoyJyorKv/bAEMBBwgICgkKFAsLFCocGBwqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKv/AABEIAZgCTQMBIgACEQEDEQH/xAAfAAABBQEBAQEBAQAAAAAAAAAAAQIDBAUGBwgJCgv/xAC1EAACAQMDAgQDBQUEBAAAAX0BAgMABBEFEiExQQYTUWEHInEUMoGRoQgjQrHBFVLR8CQzYnKCCQoWFxgZGiUmJygpKjQ1Njc4OTpDREVGR0hJSlNUVVZXWFlaY2RlZmdoaWpzdHV2d3h5eoOEhYaHiImKkpOUlZaXmJmaoqOkpaanqKmqsrO0tba3uLm6wsPExcbHyMnK0tPU1dbX2Nna4eLj5OXm5+jp6vHy8/T19vf4+fr/xAAfAQADAQEBAQEBAQEBAAAAAAAAAQIDBAUGBwgJCgv/xAC1EQACAQIEBAMEBwUEBAABAncAAQIDEQQFITEGEkFRB2FxEyIygQgUQpGhscEJIzNS8BVictEKFiQ04SXxFxgZGiYnKCkqNTY3ODk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqCg4SFhoeIiYqSk5SVlpeYmZqio6Slpqeoqaqys7S1tre4ubrCw8TFxsfIycrS09TV1tfY2dri4+Tl5ufo6ery8/T19vf4+fr/2gAMAwEAAhEDEQA/APpGiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAy/EPiPTfC+mC/1mZobcyCIMsbOSxzgYAJ7VT0LxxoXiO6a20y6kM6ru8uaF4mI9QGAzWD8X3uI/DOmPZwrPOur25jjZ9oY/NxntWDZS6tqvxa0uTxNYw6NLa2m62SOYSi55IxuwPUnHtQB32u+NdD8OTLDqd0wmYZ8qGJpWA9SFBIq3oniLTPEVqbjSbkTIpwwKlWU+6nkVxfw+ggu/Gvi2/vFV9RTUZIAW5KQqflA9uTWF4lln0T4o6uvh4eUZtFlnnSIYHmqE2HA74JoA7u++I/hrT75rSa8lklQ4byLeSVVPuygiuhsb631Kyju7OTzIZBlWxjNcr8MNPsIfh7YSWsUbG5RmmkwMyNvOcnv6V1tvbQ2kIitokijHRUGAKAJaK83t7W68d+LdZS81S+sbLTXSKGCynaFiSDksR15FWr2S70v4jeHtLS+uHt/sx3h5CfMOW5b1NAHfUV5jLq19/wmHj2Fb2byrTR1kgQOcRPiXlfQ8D8qz9M8N61e/DNtbTXtTl1uSyJhxcsI84GPkz196APXqK8xn8S3Xie58J2ek3MsQmBub5onIPlKGTt/tqK5688Ry67q+ty3kniZJLO7ltrNNKt5DCuzoWK8MeRkUAe30Vz3gbUdR1Xwja3OsW81vdZZGWaMo5CnAYg9MgZroaACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKjncxW0si9VQsM+wrhvD99471/wAP2mqRXmhxJdJvVGt5CVGf96gDvaK5P7N4+/6COg/+A0v/AMVR9m8ff9BHQf8AwGl/+KoA6yiuT+zePv8AoI6D/wCA0v8A8VR9m8ff9BHQf/AaX/4qgDrKK5P7N4+/6COg/wDgNL/8VR9m8ff9BHQf/AaX/wCKoA6yiuT+zePv+gjoP/gNL/8AFUfZvH3/AEEdB/8AAaX/AOKoA6yiuT+zePv+gjoP/gNL/wDFUfZvH3/QR0H/AMBpf/iqAOsork/s3j7/AKCOg/8AgNL/APFUfZvH3/QR0H/wGl/+KoA6yiuT+zePv+gjoP8A4DS//FUfZvH3/QR0H/wGl/8AiqAOsork/s3j7/oI6D/4DS//ABVH2bx9/wBBHQf/AAGl/wDiqAOsork/s3j7/oI6D/4DS/8AxVH2bx9/0EdB/wDAaX/4qgDrKK5P7N4+/wCgjoP/AIDS/wDxVH2bx9/0EdB/8Bpf/iqAOsork/s3j7/oI6D/AOA0v/xVH2bx9/0EdB/8Bpf/AIqgDrKK5P7N4+/6COg/+A0v/wAVR9m8ff8AQR0H/wABpf8A4qgDU8SeHYvElnaW887wC2u47oFADuKZ457c1X8Q+E4Nfm0yc3Mlrc6bN5sM0agn7pXBz2wTVP7N4+/6COg/+A0v/wAVR9m8ff8AQR0H/wABpf8A4qgBmr+ARd67JrOiavcaLfzKFmlgRXEgGeqtxnk81b8N+C7TQHurie4k1G+u8efdXAG5wO2OgHsKr/ZvH3/QR0H/AMBpf/iqPs3j7/oI6D/4DS//ABVAFKP4c3GnTSr4d8S3ulWcrFmtI4kkUZ64LZI/Cut0nThpWmxWgmkn8scySHLMfWuf+zePv+gjoP8A4DS//FUfZvH3/QR0H/wGl/8AiqAE1XwNJca5Nquh63c6Lc3CgXBgiVxLgYBIbgUa14FOrxaZKmsXFrqWnEmO+SNSzZznKnjuaX7N4+/6COg/+A0v/wAVR9m8ff8AQR0H/wABpf8A4qgCrpPw2h0y4124l1W5u59btfs9xJIgGDhssMf73TpXTaHpEeiaHbaZHI0yW8YjDuMFgBisP7N4+/6COg/+A0v/AMVR9m8ff9BHQf8AwGl/+KoAj8JfDuy8Ja5f6lbXc1w10NiRyKNsCZ3bV9s5P40t54FnGqXN5oGv3Wjfa3Mk8UMSSK7Hqfm6E+1P+zePv+gjoP8A4DS//FUfZvH3/QR0H/wGl/8AiqAOi02zewsI7eW5e5dfvSuACx9eKtVyf2bx9/0EdB/8Bpf/AIqj7N4+/wCgjoP/AIDS/wDxVAHWUVyf2bx9/wBBHQf/AAGl/wDiqPs3j7/oI6D/AOA0v/xVAHWUVyf2bx9/0EdB/wDAaX/4qj7N4+/6COg/+A0v/wAVQB1lFcn9m8ff9BHQf/AaX/4qj7N4+/6COg/+A0v/AMVQB1lFcn9m8ff9BHQf/AaX/wCKo+zePv8AoI6D/wCA0v8A8VQB1lFcn9m8ff8AQR0H/wABpf8A4qj7N4+/6COg/wDgNL/8VQB1lFcn9m8ff9BHQf8AwGl/+Ko+zePv+gjoP/gNL/8AFUAdZRXJ/ZvH3/QR0H/wGl/+Ko+zePv+gjoP/gNL/wDFUAdZRXJ/ZvH3/QR0H/wGl/8AiqPs3j7/AKCOg/8AgNL/APFUAdZRXJ/ZvH3/AEEdB/8AAaX/AOKo+zePv+gjoP8A4DS//FUAdZRXJ/ZvH3/QR0H/AMBpf/iqPs3j7/oI6D/4DS//ABVAHWUVyf2bx9/0EdB/8Bpf/iqPs3j7/oI6D/4DS/8AxVAHWUVyf2bx9/0EdB/8Bpf/AIqs7XdS8ceH9NGoXV3os0KzRRukdvIGIZwvBLe9AHe0UUUAFFFFABRRRQBFdXKWdpLcShikSF2CLkkD0Hesnw14s07xUt2dNS5jazkEcqXMJjYEjI4PtW023ad+NuOc9K4fwPLEPFfi8LImPtsWAGH/ADyFAHQa/wCKdP8ADvki9W4mlnJEcNtEZHOBknaOat6Rq9prmnJe2DlomJGGGGUjqCOxHpXLX8kcHxXtZr1lWCSy2wMx43jcWx+FP+Gozp2rSR/6h9UuTH6H5zyPagDrrv8A48Z/+ubfyrnvhx/yTjRf+vcfzNXvEem6hqFk39n6xNpuxGLeXEj7+D/eqh8NQV+Guhhm3EWwyT35NAHUUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUVXvb+106Dzr6dIY843OeKaTbshNpK7LFFY3/CXaB/0FIPzNH/CXaB/0FIPzNX7OfZke1p/zI2aKxv+Eu0D/oKQfmaP+Eu0D/oKQfmaPZz7MPa0/wCZGzRXL654u0j+wrz7HqkJn8lvL2k5zisTQPB8mr6HbX82tXqPOpYqG4HJ960jR93mm7fIzlX97lgr/M9Dorjf+Fe/9R2+/P8A+vR/wr3/AKjt9+f/ANelyUv5/wAA9pV/k/E7KiuN/wCFe/8AUdvvz/8Ar0f8K9/6jt9+f/16OSl/P+Ae0q/yfidlRXG/8K9/6jt9+f8A9ej/AIV7/wBR2+/P/wCvRyUv5/wD2lX+T8TsqK43/hXv/Udvvz/+vR/wr3/qO335/wD16OSl/P8AgHtKv8n4nZUVxv8Awr3/AKjt9+f/ANej/hXv/Udvvz/+vRyUv5/wD2lX+T8TsqK43/hXv/Udvvz/APr0f8K9/wCo7ffn/wDXo5KX8/4B7Sr/ACfidlRXG/8ACvf+o7ffn/8AXo/4V7/1Hb78/wD69HJS/n/APaVf5PxOyorjf+Fe/wDUdvvz/wDr0f8ACvf+o7ffn/8AXo5KX8/4B7Sr/J+J2VFcb/wr3/qO335//Xo/4V7/ANR2+/P/AOvRyUv5/wAA9pV/k/E7KiuN/wCFe/8AUdvvz/8Ar0f8K9/6jt9+f/16OSl/P+Ae0q/yfidlRXG/8K9/6jt9+f8A9ej/AIV7/wBR2+/P/wCvRyUv5/wD2lX+T8TsqK43/hXv/Udvvz/+vR/wr3/qO335/wD16OSl/P8AgHtKv8n4nZUVxv8Awr3/AKjt9+f/ANej/hXv/Udvvz/+vRyUv5/wD2lX+T8TsqK43/hXv/Udvvz/APr0f8K9/wCo7ffn/wDXo5KX8/4B7Sr/ACfidlRXG/8ACvf+o7ffn/8AXo/4V7/1Hb78/wD69HJS/n/APaVf5PxOyorjf+Fe/wDUdvvz/wDr0f8ACvf+o7ffn/8AXo5KX8/4B7Sr/J+J2VFcb/wr3/qO335//Xo/4V7/ANR2+/P/AOvRyUv5/wAA9pV/k/E7KiuN/wCFe/8AUdvvz/8Ar0f8K9/6jt9+f/16OSl/P+Ae0q/yfidlRXG/8K9/6jt9+f8A9eo38CRoxU6/el8ZCbuT+tHJS/n/AAD2lX+T8TtqK8Q0TRPF3iPxE4h+2abo0EmPOu3Illx6AEivaraH7NaxQ72fy1C7m6nHeonGC+F3NISm/ijb5ktcp8Sf+RLl/wCvq3/9HJXV1ynxJ/5EuX/r6t//AEclZmh1dFFFABRRRQAUUUUAR3EEV1byQXCB4pFKup7g9qwLD4feFNMuZLiw0S3gll++6bst9ea6OigDK1Hwzo2r2MVnqWnxXFvD/q0fOFq7Y2NtptnHa2MKwQRjCIg4AqxRQBDd/wDHjP8A9c2/lXPfDj/knGi/9e4/ma6G7/48Z/8Arm38q574cf8AJONF/wCvcfzNAHT0UUUAFFFFABRRRQAUUUUAFFFFABRRRQAV5l4o8QL4qki0yw0+eRYboCSTHynivTa434d/8e+q/wDX4f5V00GoJ1Lao5a6c3GneydzYHhHQdozpcHT0NL/AMIjoH/QLg/I1s0Vl7Sfdm3sqf8AKvuMb/hEdA/6BcH5Gj/hEdA/6BcH5Gtmij2k+7D2VP8AlX3GN/wiGgf9AuD8jWrBBFbQrDAgjjQYVR0FSUVLlKW7KjCMdkFFFFSUFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABSO6xoXkYKoGSScAVh+KPGOi+ELA3OtXaxZHyRA5d/ZV71g6xpt58S/DOmy2OoXejWU533MRjKSSJ/dzkEUAbF74jOqaVqC+Dpre+1G1G0KTlQ3PfPt61g+DvAOp2usDxH4v1WS/wBXZcLGpxFCDztA/wDr10/hnwnpHhLTRZ6JarCn8b4+Zz6se9bVAAAB0GKKKKACuU+JP/Ily/8AX1b/APo5K6uuU+JP/Ily/wDX1b/+jkoA6uiiigAooooAKKKKACiiigAooooAhu/+PGf/AK5t/Kue+HH/ACTjRf8Ar3H8zXQ3f/HjP/1zb+Vc98OP+ScaL/17j+ZoA6esTxVr7+HtLS5ig8+R5AioT1Nbdcf8R/8AkD2X/X0v8jWtGKlUSZjXk4020UdU1bxfqemvaw6G1sZcDzVY5UZqe2+H8rWsTXGt3yylAXAbocc967WP/Vr9BTq0+sSStBWI+rxbvNt/15HG/wDCvf8AqO335/8A16P+Fe/9R2+/P/69dlRU/WKvcf1al2ON/wCFe/8AUdvvz/8Ar0f8K9/6jt9+f/167Kij6xV7h9Wpdjjf+Fe/9R2+/P8A+vR/wr3/AKjt9+f/ANeuyoo+sVe4fVqXY43/AIV7/wBR2+/P/wCvR/wr3/qO335//XrsqKPrFXuH1al2ON/4V7/1Hb78/wD69bfh3w7B4cspLe3leXzH3s79Sa16KmVapJWbKjRpwfNFahRRRWRsFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUVma9rkGg6Jd6jLG862qbmihG5j7YFAGkzBFLMcBRkn0rzvUfii154kTRPBWnNrFxHKFuphkRQrn5snrkc0vhG/8aeKdbGr6rCmk6GQRFYuMySqR1bB46/pXcafpGn6Usg060itxI5d9g+8x6mgDDl8BaLf+Jl8Q6nbG4vdi4SRi0cZAHRScdvSunVVRQqKFUcAAYApaKACiiigAooooAK5T4k/8iXL/ANfVv/6OSurrlPiT/wAiXL/19W//AKOSgDq6KKKACiiigAooooAKKKKACiiigCG7/wCPGf8A65t/Kue+HH/JONF/69x/M10N3/x4z/8AXNv5Vz3w4/5Jxov/AF7j+ZoA6euP+I//ACB7L/r6X+RrsK4/4j/8gey/6+l/ka3w/wDFiYYj+FI66P8A1a/QU6mx/wCrX6CnVgbhRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVHcXMNpbvPcyrFFGMs7nAA+tc/4i8eaD4YuILbULtTdTttjt4+XPvisPxN4D1Hxp4gik1PWJY9BVQ62UJ2mRvRiMHGPegC94r1nX7/RrT/hXy2t2bwkNeGQFIl459+tT+CPCN14asbhtW1S41O+vGEk7yuSoPooJOBXQaXpdpo2mw2GnQiG2hUKiDsKt0AFFFFABRRRQAUUUUAFFFFABXKfEn/kS5f+vq3/APRyV1dcp8Sf+RLl/wCvq3/9HJQB1dFFFABRRRQAUUUUARXM621rJO4ysaliPpXl8Gp67b6bZeKZ9XuJI5rxkmsSR5QiDMBtGM54Hf1r1KaJZ4HikGVdSpHtXm8HhTxK1va6DcwW/wDZNtdtMbsTDfKhLHZs7deue1AGz8RdXvNM0O1mtbmW0gllAuJ4OZEXj7o5z+Rqv8OdYudTuNUQajc6hp8LILea8G2Yk53BlIBA6Y4q14l0jW9VttMvLS1hF3p8xf7I8w2SgjHLY7delSeFdG1SHxBq2ua1bw2c1+sSC2hkEgQICM7uM5zQBp+I/ENhoVk39oNMPNRgvlwPJ2P90HFUPhqwf4a6Gy9DbAjI9zXRXf8Ax4z/APXNv5Vz3w4/5Jxov/XuP5mgDp64/wCI/wDyB7L/AK+l/ka7CuP+I/8AyB7L/r6X+RrfD/xYmGI/hSOuj/1a/QU6mx/6tfoKdWBuFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUVzPjXxnD4Q09JPsVxfXU7bLe3gjLF27DjpQB0F5eW9haSXV5KsMMS7ndzgAVxWhfEdfFniltO0HTbibTI0bzdSI2pnsFzyak8M2fiDxNol6PH9rbLa3jBobJVBKJwQGPc5HpXW6dptnpFjHZ6bbR21vGMLHGoUD8BQBzXh74baJoOrXGqsr32ozyM/2m5O5kBOdo9v8K6+iigAooooAKKKKACiiigAooooAKKKKACuU+JP/ACJcv/X1b/8Ao5K6uuU+JP8AyJcv/X1b/wDo5KAOrooooAKKKKACio3nhjkWOSVFd/uqzAFvoKerq+djBsHBwc0ALRRRQAUUUUAQ3f8Ax4z/APXNv5Vz3w4/5Jxov/XuP5muhu/+PGf/AK5t/Kue+HH/ACTjRf8Ar3H8zQB09cf8R/8AkD2X/X0v8jXYVx/xH/5A9l/19L/I1vh/4sTDEfwpHXR/6tfoKdTY/wDVr9BTqwNwooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooozjrQAVDc3lvZx77qZIlJABc45rIPjLRG8RroMN6kuoOhYRxgsBjHUjgda4+H4b6r4j8QnVPH2p/a4YJd9rYQZWJMdCQev50AWvGV1421jXD4f8L2v9n2TIGm1WT0PVV6811mg6ENJ0S1sryc6hNBz9onUFix5z7VrKoRQqjAAwKWgAooooAKKKKACiiigAooooAKKKKACiiigAooooAK5T4k/8iXL/wBfVv8A+jkrq65T4k/8iXL/ANfVv/6OSgDq6KKKACiimTGQQP5AUy7TsDHAz70AeS+LZLnS/iLa6jqEF/NCmoxSQNBGzosAiIYcdDu7V2ngY3U1tqd9cxTQw3t801rHOCrJGVUAFT05B4rlBc+N9K1vVp9Qj0AR3U6NEt1cybVATHy/L+fvXXeB47n+z725vr2C6mu7ppmFu5aOLIA2KT24/WgDpqKKKACiiigCG7/48Z/+ubfyrnvhx/yTjRf+vcfzNdDd/wDHjP8A9c2/lXPfDj/knGi/9e4/maAOnrj/AIj/APIHsv8Ar6X+RrsK4/4j/wDIHsv+vpf5Gt8P/FiYYj+FI66P/Vr9BTqbH/q1+gp1YG4UUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRXCeNPiONC1EaHoNhJq2vSKCltGPlTPQse3UUAdVrWvab4e0973WLuO2gUfedgM+w9TXL6V4jPxL0PUYdNivdLtWG23vWVkMgOeV6fzq4nhceK9D0uTxzZRSX1s3mtFG7bA3PUcZ4PeupggitoVit4kijUYVEUAD8BQBzfhD4f6L4OgJsYfOvJOZbyb5pHPf5jz1rqKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAK5T4k/8AIly/9fVv/wCjkrq65T4k/wDIly/9fVv/AOjkoA6uiiigAooooA8p8UeEptU8SX2qvYLqv2S/ilW2EoDNCIsMnXjLHPNdN8ONJn0nSL4S6cdMguLxpba0LhjFGVAAJHuDUF54X8UQeKtQ1Pw/rdraW98VZ4ZbQSHKqFznI9K6TQoNYt7Fl1+9hvLjeSskUPlgL6YyaANOikZgilmIAAySe1cPbfEK5mureeXSPL0a5uTbRXvnZbcCQSUxwMqec0AdzRWR4i8QR6Bpi3HlGeaVgkMKnmRicf1pdCvdZvI5P7c0uLT3XGwR3Hm7vxwMUAaF3/x4z/8AXNv5Vz3w4/5Jxov/AF7j+Zrobv8A48Z/+ubfyrnvhx/yTjRf+vcfzNAHT1x/xH/5A9l/19L/ACNdhXH/ABH/AOQPZf8AX0v8jW+H/ixMMR/CkddH/q1+gp1Nj/1a/QU6sDcKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKRmCKWcgKBkk9qAFqKW5iibY0ieYVJWPcAzfQVz914nXV9K1WPwXc2t/qdmNoTzBtDkcAn8Kw/B/gDUrbWB4i8X6tNfaseUiRyIYP8AZC5waAM6W98c+N/ERg01H8O6NZ3ADzvzLPtPIGOx+nevSYbC3imE5jVrjaFaYj5mwMdas0UAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABXKfEn/AJEuX/r6t/8A0cldXXKfEn/kS5f+vq3/APRyUAdXRRRQAUUUUAcJqXjXUYfHVvpVrbwCxW+js55HYhyzxlwQOmMCug8O65LrM2qxyxon2C9a2UochwFBz+tcX450yB/EBvdS0i9htYXWUajZyqOQuMsvJJHI6V1fgW58P3Ph4N4Wk821VyruVIYvjndnv0oA3b2A3NjNCpwZEKg+ma8jj1GG78L2HhZAy6st84lg2nMS7mO8+2MfnXsdM8mMPvEa7v720ZoA89+IkMOsaLpV9b3DGys7jNxJFnIUgA1N4BmibxDq0ehXEtzoKpEbeSR2cbyDvAZuTziu98tCu3Yu30xxQkaRjEaKo9FGKAMjxH4esNdsm/tBZj5SMV8ud4+x/ukZqh8NVCfDXQ1XoLYAZPua6K7/AOPGf/rm38q574cf8k40X/r3H8zQB09cf8R/+QPZf9fS/wAjXYVx/wAR/wDkD2X/AF9L/I1vh/4sTDEfwpHXR/6tfoKdTY/9Wv0FOrA3CiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKCQASegrznVfijJdeIF0PwVpsmrXaybZ58YihwcEEnGT+NAHU+J/GGjeEdPNzrF2sRP+riBy0h9AKwdU0+5+J/hTTprO+vdEs5zvni2gSSLxge3foa1J/Aeiaj4ki8QapaC4vkjAVZDlYz6getdMqhVAUAAdABQBj+GvCuleE9NFlo1ssSdXc8s59STya2aKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAK5T4k/8iXL/ANfVv/6OSurrlPiT/wAiXL/19W//AKOSgDq6KKKACorq4W0tJbh0kdYlLFY13McdgO5qWo7l3jtZHj2b1Uld5wuff2oA818S39pceIpl1VfFEtsFRkg0+zkEeCoOCy8N9K63wW+kvoz/ANh6Xc6bAJSHS6tjA7tgfMQev1rjrPU/Hljr+svLDoO2eZGjWe7lAVQgHy/L09feuv8ABsOoJZ3txq9zbzXF1dNMVtpC8cWQBtUnnHH60AdHRRRQAUUUUAQ3f/HjP/1zb+Vc98OP+ScaL/17j+Zrobv/AI8Z/wDrm38q574cf8k40X/r3H8zQB09cf8AEf8A5A9l/wBfS/yNdhXH/Ef/AJA9l/19L/I1vh/4sTDEfwpHXR/6tfoKdTY/9Wv0FOrA3CiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiio7i4htYHmuZFijQFmZjgACgCSs/WtWj0XRrrUHiluBbrkxQqWdj0AAH1rnvFOsa9qGhwN8Phb3TXT+WbtmO2HsW6dqk8D+DJvDNrcSapqc2p3162+4eU5UHjhR0xx6UAY/hG+8beKNeXWdVhj0nQgjCKxdMyyZHVuhHbtXcado+n6Ssi6bZw2wlcvJ5SBdzE5JOPrV0AAYAwKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigArlPiT/yJcv/AF9W/wD6OSurrlPiT/yJcv8A19W//o5KAOrooooAKjngiureSC4QPFIpV1PQg9qkooA8t8RN4ck1LWY5tBtbnUraWK2tlkY7p2MYI4z0A4/Cuk+HjwLpV9aQ6dDYS2d20M8cOdjOFBJGSexFZGqxXGreOLp9A0azkutLZRNdTsAfMKgrgEeh61v+Cb1by11HzLNbS9ivGS9RG3Ay7Rk5xzxigDpqKKKACiiigCG7/wCPGf8A65t/Kue+HH/JONF/69x/M10N3/x4z/8AXNv5Vz3w4/5Jxov/AF7j+ZoA6euP+I//ACB7L/r6X+RrsK4/4j/8gey/6+l/ka3w/wDFiYYj+FI66P8A1a/QU6mx/wCrX6CnVgbhRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFQXl7b6faSXV7MkEEYy7u2AB9a4nQfiV/wlfiwWPh3TJrnSYwwm1JsqgYEDC8YPfv2oA2PEnj3QvC80NvfXPmXU7hI7eEF3JPsM4/GsbxT4C1Dxr4hRtT1mRPDyorCwhypkcHncT26VoaB8NtD0PWLjVjHJe6hO5c3F23mMn+6T0rrqAKelaVZaJpsVhpkC29tEMKi9BVyiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACuU+JP/Ily/8AX1b/APo5K6uuU+JP/Ily/wDX1b/+jkoA6uiiigAooooA8u8VwJqXiy7TQNPv5L6AKt7NbuQmSAQDz1xjtXUfD+TTzoUsOnWs9rJBOY7qK4OZBLgZ3HJ5xiqeo+CdVl8SXuqaRr82nrebTJDH0JCgZPHXineF/Bmq+HtQaabXZbuKWRpZ0f8A5auRjJ4+n5UAdnTBNGX2CRS393cM1Hezm2sZplGTGhYD1xXkkVhHZ+GdP8VRsx1dr52ln3cyruYBD7Yx+VAHsVIrq4yjBgDjg5rmvGPiAaTpNvbxyxxXeot5MRkcKBx8xyeOASawPhCILfStUs4b1Lpor+YFxIGLfMfm/GgDv7v/AI8Z/wDrm38q574cf8k40X/r3H8zV7xHqWoafZN/Z+jzalvRg3lyomzg/wB6qHw1Jb4a6GWXaTbDIPbk0AdRXH/Ef/kD2X/X0v8AI12Fcf8AEf8A5A9l/wBfS/yNb4f+LEwxH8KR10f+rX6CnU2P/Vr9BTqwNwooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooqC6vrWxjV7y4igVmCgyOFyT25oAnrmPGvjSDwfpsMptZr25un8q2ghXJd8Zx+lYvjO58a6xrQ0LwnEun2boGl1dmBxkD7oBz39O1dX4f0M6PoNpYXdzJqEtuOZ58FiSSSenvQBz/AIVt/EniLS7x/HtpBDb3g/dWCk5iHoxxXWadptnpNmlrp1tHbwoAAkagVaooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACuU+JP/Ily/wDX1b/+jkrq65T4k/8AIly/9fVv/wCjkoA6uiiigAooooA891TxdrEfxAttOtZoYbBNQjs5o3jDNKHjL7g3bGMV0fhbW59Zk1cTsjpZ3zW8LoMB0Cqc/qa5bxt4eutb1ou3hqO9jgIaKdbmSJmOOvykc9q3Ph5e/adCntzpyaa9ncGB4FYsVIAPzE8k80AdWyh1KsMgjBB71w1v8PbqK5gt5NY8zR7a4a5iszD8wYknBfPIyx4xXdUUAZE/h601W0ii8R21pqjRMShltxhfoDmqHhHwTp/hE3xso4A11cPLmOEJtUkkL9BnFdNRQBDd/wDHjP8A9c2/lXPfDj/knGi/9e4/ma6G7/48Z/8Arm38q574cf8AJONF/wCvcfzNAHT1x/xH/wCQPZf9fS/yNdhXH/Ef/kD2X/X0v8jW+H/ixMMR/CkddH/q1+gp1Nj/ANWv0FOrA3CiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKz9a1zTvD2mvfavdR21ug5Zz19q5XSfEbfEzQdWg0+G90q1YeVb35QAvnILLz2oA2pPGuif8JEmgQXyS6lIDiKMhth/wBr0rlF+G2p+I/Esmp+OtUa6toZt1pYQHbGoByrEZwT+Hauj8I+ANF8HwH7FF592/Mt3N80jn1yc4/CuooAbGixRqiDCqAAB2FOoooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigArlPiT/wAiXL/19W//AKOSurrlPiT/AMiXL/19W/8A6OSgDq6KKKACiisTxpcTWngfWbi2dkmispWRkOCCFOCDQByt8I9W8Y6vazeMdU0VrV0CwR3cSIwKA5UEZrqvCuk6fpGnSppl6b8zSmSe5aRXaSQgAlivGcAV5+htNWgtr/U/AzXl3Jbx77iSLcz/ACjnO2u68GxW8WkSLa6N/ZCeacwbdu44HzdBQB0NFFFABRRRQBDd/wDHjP8A9c2/lXPfDj/knGi/9e4/ma6G7/48Z/8Arm38q574cf8AJONF/wCvcfzNAHT1x/xH/wCQPZf9fS/yNdhXH/Ef/kD2X/X0v8jW+H/ixMMR/CkddH/q1+gp1Nj/ANWv0FOrA3CiiigAooooAKKKKACiiigAooooAKKKKACiio5LiKORY2dRIwJVM8tj0oAkrhfGvxF/sC+TRdE06fU9anH7uJEIVP8AaYnjH41j3F3468b6+9tp8beHtFtZisk8inzZypx8o4ODg/pXpMFhBFIs5iRrkRhGm2/Mw+tAHNweGD4q8P6YfH9jbz39sfNMSZ2K3Hv7V1MEEVtCsUEaxxoMKqjAAqSigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAK5T4k/wDIly/9fVv/AOjkrq65T4k/8iXL/wBfVv8A+jkoA6uiiigAqvf2UGpafPZXa74LiMxyLnGVIwasUUARwQJbW0UEIxHEgRR6ADAqSiigAooooAKKKKAIbv8A48Z/+ubfyrnvhx/yTjRf+vcfzNdDd/8AHjP/ANc2/lXPfDj/AJJxov8A17j+ZoA6euP+I/8AyB7L/r6X+RrsK4/4j/8AIHsv+vpf5Gt8P/FiYYj+FI66P/Vr9BTqbH/q1+gp1YG4UUUUAFFFFABRRRQAUUUUAFFFFABSMyopZyFA6kmsTxP4w0fwlYNc6vdKh/giX5nc9gFHNYGpWNx8UPCdpJa3l5otpO/7+L7ryR9x04yKANebxTDrGn6pB4Qura91SzG0Rsx2q/HU49DWD4P+H2o22tDxJ4w1WW+1gqQsSORDDkc4Axnr3FdT4a8KaT4T00WejWqwr1dh96Q+pPrWzQAYx0ooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigArlPiT/AMiXL/19W/8A6OSurrlPiT/yJcv/AF9W/wD6OSgDq6KKKACiiigAooooAZNKsELyyHCICxPtXnFv4s8SLDaa/dz250i5u2ha0EIDxoCw3b+/3emO9ei3MAubWSBzhZFKn8a8uh03Xp9Ls/C0+j3KRwXjPNfEDymiLMRtOc55Hb1oA9C1mHV7q1iXQbyGzkY5aWWHzAB9MisnwfrepX17qem6vLFdT2DqPtMKbFkznjHYjFReLNa1y28LqfD+h3s93M3lFFUbol4y3X0zipfAUItdHe2Ok39jIrb5Zb5FDTuercE9/wCdAGvres6bpNlJ/ad9b2nmRsE86QLu47ZrK+GzBvhtojKcg2wII78mtzU7K1u7KX7VbQz7Y2x5kYbHHvWJ8NwB8N9EAGB9mHA+poA6euP+I/8AyB7L/r6X+RrsK4/4j/8AIHsv+vpf5Gt8P/FiYYj+FI66P/Vr9BTqbH/q1+gp1YG4UUUUAFFFFABRRRQAUUVQ1jVk0nSbm98p7jyELGKHBY/hQBfrzfV/ijLea6NE8C6e2r3aSBLifpDD65bBFL4TvfG/inxBHrWpRjRtEVSIrFlBkmB6M2RxwOx713ljpVjpu/7DaQwGQ7nMaAFj70AYlz4E0TU/E0fiDVbT7RfJGEVJW3xoRnkKe/PWulUBVAUYAGABS0UAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABXKfEn/AJEuX/r6t/8A0cldXXKfEn/kS5f+vq3/APRyUAdXRRRQAUUUUAFFFFABRRRQAUUUUAQ3f/HjP/1zb+Vc98OP+ScaL/17j+Zrobv/AI8Z/wDrm38q574cf8k40X/r3H8zQB09cf8AEf8A5A9l/wBfS/yNdhXH/Ef/AJA9l/19L/I1vh/4sTDEfwpHXR/6tfoKdTY/9Wv0FOrA3CiiigAooooAKiuLmG0gaa6lSGJRlndgAPxNc94m8faF4Wngtr+58y7nkVEtoRvkOTjO0c4rH8T+Bb3xxrcMmo6vNF4f8tSdPiBRpDjncc/pigC74o1jXdS0G2k+H6wXTXchja5ZhthXnLdRnkU/wP4MuPDMM8+rarNqmo3nNxLKflz7DHFdFpOk2Wh6bDYaXbpb20K4REGAKuUAAAAwBgUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVynxJ/5EuX/r6t/wD0cldXXKfEn/kS5f8Ar6t//RyUAdXRRRQAUUUUAFFFFABRRRQAUUUUAQ3f/HjP/wBc2/lXPfDj/knGi/8AXuP5muhu/wDjxn/65t/Kue+HH/JONF/69x/M0AdPXH/Ef/kD2X/X0v8AI12Fcf8AEf8A5A9l/wBfS/yNb4f+LEwxH8KR10f+rX6CnU2P/Vr9BTqwNwoormvGnjGHwhpUdx9jnvri4by7eGBCxdvcgYA570Ab93eW9haSXV5KsMESlnduij1rh9G+JJ8V+KP7P8OaZNc6WhKXGpEYRT/s881P4VtfEfiLS79vHttbLaX4AisADlE54Y57giuu0/TbPSrNLXTreO3gjGFRBgCgDnNE+HOh6NrV1qxia8vrhy3nXJLlB6KDnFdZRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUjMq/eYD6mm+dH/AM9E/wC+hQA+imedH/z0T/voUedH/wA9E/76FAD6KZ50f/PRP++hR50f/PRP++hQA+imedH/AM9E/wC+hR50f/PRP++hQA+imedH/wA9E/76FHnR/wDPRP8AvoUAPopnnR/89E/76FHnR/8APRP++hQA+imedH/z0T/voUedH/z0T/voUAPopnnR/wDPRP8AvoUedH/z0T/voUAPopnnR/8APRP++hR50f8Az0T/AL6FAD6KZ50f/PRP++hR50f/AD0T/voUAPopnnR/89E/76FHnR/89E/76FAD6KZ50f8Az0T/AL6FHnR/89E/76FAD6KZ50f/AD0T/voUedH/AM9E/wC+hQA+uU+JP/Ily/8AX1b/APo5K6kSxk4DqT7GuW+JP/Ily/8AX1b/APo5KAOrooooAKKKKACiiigAoorGh8XaDPrB0qHVIHvgceQCc59PSgDZoqrqOp2WkWT3epXCW9un3pH6Co9J1vTddtPtWkXcd1DnG9PWgCxd/wDHjP8A9c2/lXPfDj/knGi/9e4/ma6G7/48Z/8Arm38q574cf8AJONF/wCvcfzNAHT1x/xH/wCQPZf9fS/yNdhXH/Ef/kD2f/X0v8jW+H/ixMMR/CkddH/q1+gqK8vrXT4fOvriO3jzjfIwUZ/GsQ+NtDXxDB4fjvo5NSlU4iTnbgZ5/KuXHw11LxF4mk1PxxqjXNtDMTaWMDFYwuflLYxk4xWBuWvGVx401fVY9F8KRR2dlMm6bU2bOF/2R6/jXUeHNA/sPQrWwubuW/lgHM9wxZifXmtaONYokjjGFRQqj0Ap1ABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAVZEWa/CSDcoj3AHpnNSfZLf/nin/fNRuGN84Q4byOD75rmZNO8amRjHqtqFzwNo/wDia0hBS6pGc5uPRv0Oq+yW/wDzxT/vmj7Jb/8APFP++a5P+zfHH/QWtf8Avkf/ABNH9m+OP+gta/8AfI/+Jq/Yr+ZGftn/ACM6z7Jb/wDPFP8Avmj7Jb/88U/75rlF03xsGG7VbUjPPyj/AOJrroQ4hQSnMgUbiO5xzUTgo9UzSE3Lo16lWdtNtZ4IbgwRSXLlIUfAMjAZwPU4GabFNpc97LZwvbvcwgGSJSCyA9Mjt0rz34z3eo2N54SuNGh867TUZdi+3kPk/gMn8KVdYs/C/wAM01rQjJqF7fTxRSzlQZWkkYKeDxkE9OlZmh6V9kt/+eKf980fZLf/AJ4p/wB815lZa/4wtVv/ALZb6n9lWyeZbq/t44/LkBGFUITkHJ6+lW4NS1iz8Ex6rrHiG4NzfKBDDBbxko27omRycDvQB6F9kt/+eKf980fZLf8A54p/3zXmOieLPE1x/wAJBpJivZb/AE+3jmia7hRJiH3fwr8vG3iug8CatLfGaG/1u4vbxVBltbmBI3hOOR8vWgDqZv7Pt5Ikn8iN5m2xq2AXOM4HrUv2S3/54p/3zXKag3234qWVpP8A6qysvtcef75ZkP6Vy2veM9WsNZe80u61K7sI79baQfZohbr8+GXf97IoA9U+yW//ADxT/vmj7Jb/APPFP++a4W01DXtZ8cazFFqzWum2AgdYkiVid0QYrkjoc1z+leLvGOrLa6tY2WrTwzS4+z/Z4vs+zdtJ353epoA9a+yW/wDzxT/vmj7Jb/8APFP++a8w8V+L9UsNUu7rRrzUrm2spds0cVtEYFIPzKzn5q9Jtrwz6LDe4VTJAsuGPAyoP9aAJvslv/zxT/vmj7Jb/wDPFP8AvmvM5PihqCSMovPDPBI5uJ8/+gV6JpN69/otteOYXeWIOTASUJI7E84oArnVdAGqDTTeWQvScC33rv8Ay61fe3tY0Z3ijVVGSSBgCvIEghPwnvdbKq2qx307pPj95uWVlUZ69O1el69ZXWqeEJra0fbcSQqQc4z0JH49KAJdP1PQdVmeLTLuyu5I/vrC6sV+uKYNZ8OnUhp4vrE3hO3yPMXfn0xXnVjdXVj4w8MwJoUukGK3ljuGkQL52AoJGD8wHqfWseG2vYdIivHt7ZtETVJHOoA/6Xje+e2MA8DnpigD1y41vw5aX32K5v7CK6zjyXkUNn6VfP8AZ63SW58gTOCyx8bmA6kCuN8bLpf2FNPs7GK51fUmXysIC46EsT1GACfwo8RQS6EPC9x5hkureaOzZ8/fDlVY/pQB2N3bxRWryRxqjKMgqMVz/wASP+RJf/r5tv8A0cldJff8eMv+7XN/Ef8A5Eh/+vm2/wDRyUAdZRRRQAUUUUAFFFFAFe/aRNPnaDPmCMlceuK8oWK3j+G+lXNuF+2LqMjRuPvGTe//ANevX65eD4faJb6st9H9qykhlS3M5MKOerBOgPJ596AKPjfEureFobnm1kun84N0JCZXP/AsU7w6qw/E/wATw2oC24gtW2r90MQ2ce9bd/4WsNT0j+zrxriSPcGWQynzEPqG6jpT9B8N2Ph2CSOxM0jStukmuJDJI/1Y8mgBniOPXJLJv7CuLKHCN5n2qNmyMHpgiqHw13f8K10Pfgt9mGcdOprorv8A48Z/+ubfyrnvhx/yTjRf+vcfzNAHT1geL/DbeJ9HWyS4+zkPu3jqOMVv0VUZOLuiZRUlys5fwh8P9F8G25+wQ+bduP3t3LzJJ9TXUUUVJQUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBX/AOYkf+uX9asVBNAzyCSJ9jgYyRnIpvlXf/Pwv/fugCzRVbyrv/n4X/v3R5V3/wA/C/8AfugCzRVbyrv/AJ+F/wC/dHlXf/Pwv/fugCnq3h611jU9Jvrl5BJpc7TwqpGGZkKEH2wxrMm8AaTLcX77pkgviHe2Ujy1kBJ8wDHDZOfwFb/lXf8Az8L/AN+6PKu/+fhf+/dAGDbeCvKt7iC71zU76GaEwhLiRSEU46YA9Knv/B1jf+HrfSWmniW1OYZ4yBIhz1Bxitfyrv8A5+F/790eVd/8/C/9+6AOa074e22m3mpXsWrai97qMUccty7qXGzJBHHX5jV7QfCUGiX019JfXeoXco2me6YFgPQYArX8q7/5+F/790eVd/8APwv/AH7oAyNW0SebxRpurWe3MWYrkH+KPBwB/wACNZN58MNOvJZQ+pagtpLdfazZq6+UJM5zjGevvXW+Vd/8/C/9+6PKu/8An4X/AL90AUtM8O2ml6lf3sDyNJfCMSByMDYmwY/Csq28CRWN1u0/WdStbQPvFjFIohHtjGcfjXReVd/8/C/9+6PKu/8An4X/AL90Acrqfw00/U5rzdqN/Ba3rmSa0idRGXPJbGM5/GuttLZLOxgtY8lIY1jUt1IAx/SmeVd/8/C/9+6PKu/+fhf+/dAEn2aD/nhH/wB8CpAAqgKAAOgFV/Ku/wDn4X/v3R5V3/z8L/37oA5s/DzTjqn2gXl4LXzvPOnhh5Bf1xjPc9+9dA2nMzzkXtwqypsVAwxF7rx1qTyrv/n4X/v3R5V3/wA/C/8AfugDF0jwdBpuo/b7zUb3VblUKRyXrhjGp6gYA61Rj+G2nx3AA1C+NgsxnGnl18ncSSeMZ6k966jyrv8A5+F/790eVd/8/C/9+6AOWufh3HN4luNbt9e1W0uJgF2wyKFRR/CMqeKv6joFxf6zpHnStLZ2IZpGkOWkcY2n8xW15V3/AM/C/wDfujyrv/n4X/v3QAt9/wAeMv8Au1zfxH/5Eh/+vm2/9HJXRNbTyjbNOGTuAmM1zvxJ48FSf9fNv/6OSgDrKKKKACiiigAooooAKKKKACiiigBksfmwvGTjepXPpmuM03wLrOkabDYWPjO9jt4F2xp9kiOB+IrtqKAOT/4RfxF/0O17/wCAcP8AhWT4o03xPoXhm91K38ZXcslugZUaziwfmA9Peu21XVrPRdPe81GXyoEOCcEkn0AHJPtVPS9b0jxZYTfYn+0QghZY5I2QjvypANAGVH4Z8RvGrHxte8gH/jzh/wAKX/hF/EX/AEO17/4Bw/4Ve1Xxpoeiailhf3TJOwHypEzhcnAyQMD8a3I5EljWSNgyMMqwPBFAHK/8Iv4i/wCh2vf/AADh/wAKP+EX8Rf9Dte/+AcP+FdZRQByf/CL+Iv+h2vf/AOH/Cj/AIRfxF/0O17/AOAcP+FdZRQByf8Awi/iL/odr3/wDh/wo/4RfxF/0O17/wCAcP8AhXWUUAcn/wAIv4i/6Ha9/wDAOH/Cj/hF/EX/AEO17/4Bw/4V1lFAHJ/8Iv4i/wCh2vf/AADh/wAKxdEsfFGp61rlnL4xu40026WGNhaRZcGNWyePU103/Cb6D/bQ0oXjfaS/l/6ptm7pt34259s1c1LUdM8OW8t9dgxLM4LtFEzs7Yx0UE9BQBj/APCL+Iv+h2vf/AOH/Cj/AIRfxF/0O17/AOAcP+FbGieI9N8QwvJpc7SeWcOroUZfqrAGtSgDk/8AhF/EX/Q7Xv8A4Bw/4Uf8Iv4i/wCh2vf/AADh/wAK6yigDk/+EX8Rf9Dte/8AgHD/AIUf8Iv4i/6Ha9/8A4f8K6yigDk/+EX8Rf8AQ7Xv/gHD/hR/wi/iL/odr3/wDh/wrrKKAOT/AOEX8Rf9Dte/+AcP+FH/AAi/iL/odr3/AMA4f8K6yszXPEOneHbZJ9VmaNZG2oEjZ2Y+yqCTQByGv6b4n0kab5PjG7k+138Vq2bOL5VfOT09q1v+EX8Rf9Dte/8AgHD/AIVt6ff6b4i0+G8s2W4gWTchKkFXX2PIIqLVPE+k6Pqljp2oXQjur99lvGFJLHBPbp0PWgDJ/wCEX8Rf9Dte/wDgHD/hR/wi/iL/AKHa9/8AAOH/AArrKKAOT/4RfxF/0O17/wCAcP8AhR/wi/iL/odr3/wDh/wrrKKAOT/4RfxF/wBDte/+AcP+FH/CL+Iv+h2vf/AOH/CusooA5P8A4RfxF/0O17/4Bw/4Uf8ACL+Iv+h2vf8AwDh/wrrKKAOT/wCEX8Rf9Dte/wDgHD/hR/wi/iL/AKHa9/8AAOH/AArq3dY0Z3OFUZJPYVhaZ410LV9UOn2F2zz87d0TKr467WIw34UAct4TsvFGv6PJeXHjC7hdbqeAKtpF0SQqD07gVt/8Iv4i/wCh2vf/AADh/wAK2b/UNL8Maa09yRbW+8nCKWLMTk4A5JPNO0bXLLXrRrnTmkaNW2nzImjOfowBoAxP+EX8Rf8AQ7Xv/gHD/hR/wi/iL/odr3/wDh/wrrKKAOT/AOEX8Rf9Dte/+AcP+FH/AAi/iL/odr3/AMA4f8K6yigDk/8AhF/EX/Q7Xv8A4Bw/4Uf8Iv4i/wCh2vf/AADh/wAK6yigDk/+EX8Rf9Dte/8AgHD/AIUf8Iv4i/6Ha9/8A4f8K6yigDk/+EX8Rf8AQ7Xv/gHD/hWRf6f4otPFWk6WnjG7aO+SZnc2kWV2KCO3fNdTrni3SPDs0UWqXDpJKMqkcTSHHqQoOB7mr0M9hfW8GqRtHJGsZeKb0Ujk/lQBz/8Awi/iL/odr3/wDh/wo/4RfxF/0O17/wCAcP8AhVqw8daBqWprYWt05nZiib4XVWIOMBiMH8K6KgDk/wDhF/EX/Q7Xv/gHD/hR/wAIv4i/6Ha9/wDAOH/CusooA5P/AIRfxF/0O17/AOAcP+FH/CL+Iv8Aodr3/wAA4f8ACusooA5P/hF/EX/Q7Xv/AIBw/wCFH/CL+Iv+h2vf/AOH/CusooA5P/hF/EX/AEO17/4Bw/4Uf8Iv4i/6Ha9/8A4f8K6ys7W9dsPD2nm91SR44AduUjZzn6KCaAOc1Dw/4ktNNublPGl4zQwvIAbOHkgE+lVvD2keJtX8N6fqM3jO8jkurdJWQWcOFJGcdK7DTdRste0eG+sX86zukyjFSNynjoay73xf4f0C+j0me4MMihVCJEzLGD0yQML+NAFT/hF/EX/Q7Xv/AIBw/wCFVr/wLq+q2622o+MLye381JGjNrEN21gwGQPUV2iOskauhDKwBBHcUtABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAHGfED/XaF5/8Ax7f2lDuz03bvl/rTdF2f8LQ1n7JjyvKXz9vTzMLjPvtrqdV0mz1rT3stRi82BjkjJBB9QRyD71n6d4P0bSrXyLKCRF8wSFjM5ZmAxyxOTQByNv5H/Fefbtvn75tm/r5ez5Mf8Cziut8E+Z/wgui+dnf9iiznr9wdaTVfBWh6zqKX1/as06gDKSsgbByMgHB/GtyONIo1jjUKijCqBwBQA6iiigAooooAKKKKACo7nf8AZZfL+/sO3HripKKAPII/I/4VbN5e37V/bMuz+953mNj8c5r0u6v7XStAjvtVZUSCJWLP2bGPzqn/AMIPoP8AbQ1UWbC5D+Z/rW2buu7Znbn3xTtU8GaLrWlnT9RgkltjJ5hTz3HzfUH9KAMzwVY3N1qeoeJ7yFrU6kAsVuwwVjByCw7Nya7GsfQ/C+meHTIdLSZfMADCSd5OB6bicVsUAFFFFABRRRQAUUUUAFcbrX/JT9A+1f6ry5/J3dN3l8/0rsqzNc8O6b4itkg1WFpFjbcjJIyMp9mUgigDnfh+UFxrxtv+PX7a3l7em7c2/H415zrGvGXxVpGp6tperJfNqhCq9k+2OMIw2Ie/Pzcetey6f4Z0vS47dLGBoltySgEjdScknnk/WrGo6PZarLaSX0PmPZy+bCdxG1sEZ/ImgC5G/mRq4BAIzhhg06iigAooooAKKKKACiiigDL8S+b/AMIzf+Rnf5LdPTv+leeQ+R/wi3gT+zcedhPK29dmF3/pXqzoskbI4yrDBB7iufsfAugabfSXdpaMkrqy8zOVQMMHaCcL+GKAOd8fJqc/jTw7Bp8kcSMzESTJvVJMja23uetbPg3XL7UbnV9O1VopbnTLnyGniXasvyq2cdvvdK1L/wANaZqWlpp95C7wI25cSsGU+zA5/WpdG0LT9AtDb6ZD5SM25izFmY+pJ5NAGjRRRQAUUUUAFFFFABRRRQBwn7v/AIWTrX23G7+yfk3f889x/wDr03wrJFF8KbyS4Vns1jmKKneHbwB+FdHrnhLSPEU0cuqW7vJGMK8crRkj0JUjI9jVm30HTrVQsEG1BCIRHuO3YBjG3pQB5Xp1nqFjL4TutSvobzTZtQUWlrHGFeFm3FWLZ+bAyMYFey1zun+BNA0zUlvrS0cToxZN8zsqEnOQpOB+FdFQAUUUUAFFFFABRRRQAVT1aNJNKuN6hsISMjocVcrP1rQ7LxBp5stTWR4CckRytGfzUg0AY3w0/wCSbaN/1wP/AKEa5U+R/YHjr7bt+04m3bvvbPm2f1xXY6P4F0LQbVbfTIJ4okbcqG5kYA/iafqfgrQtX1QahfWjNcDGSsrKrY6blBwfxoAteF/M/wCEW03zvvfZo+vptGK1aREWONUQBVUAADsKWgAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAPPvBnibxF4jhtL661PQkgmkkDWiK3nhVZlx97rxnpXRf8Jron9sHTHuJY7jfs/eQOqFvTcRj9awvh14Vs9P8G2d1Po0MGrqZmMkkAWUMZHwScZ6Yrg/EVlrGpeQJ7TXZtWj1i3aVVEn2VIxKhJUdCMZ5oA9Ph8Xpb6t4gTVmjhs9LMOx1UljvQseO547VdsfGGjahptxfwTyLb25xI0sDx/kGAz17VyV/paSah4sbVrK/e0nktTG1pGxkJEbAsmOuM9qxp7DxD4i8K6nalb6Sxt5VNsJ4mgnkQbcjnk9zn2oA9J0XxXpPiCaSHTZ3aWMAsksTRtg98MBmm654v0bw7eQWmq3LRz3EbSRRpEzllUgHGAfUVzPgvT9IbxE99ZW/iEXSxbWl1XzNuOeBu696qeN75dO+Lvhm6ks5rtY7C5JSCMu6/PH8wUdSKAOwHjLQzof9rC8/0XOM7Dvz6bcZz7YpsHjXQrjSZtRS8K28DbZN8bK6nsNpGefpXnA0LVDdf8JN/Z0/2P+2/7Q+w+WfM8ryPKxs653c4q/Pp0et6lr+vajpeox6XdRW8UUEcLLP5kbs3mbOv8QP4UAdfL410668P3mo6ddGFLXAeS6tpFVcgnoQCenas+z+JNhP4qvNEl8xZIFGyRbdyrHaD1xwOa5m4Gu6r4K160EF9c2ihFsvtNs0c7fKdwIPJ571sWX2jRvF16tzYXkiahZgRSQwF0UhQMMR06UAdPpHiiyvfssEt5FLcXbOIfLjZVbYAT19Aamn8VaRbLetLdcWTKk2EJwzAkAY+8eDwK8+l0rVrH4eWuoWNjMdU029MscOw72jJXeAOvKjFMu/CeoSeBbC6uI7wzNqY1C7it2KTiMlmAGOdyhsYoA9H0TxJpniBZDpkzOYjh1kjaNh+DAGq+q+MtF0bUhp99cOt0VDCJIWckHoeAaxPAun6VHqd3e6dBriTPGI5JNW35IyDgbqmGmzN8XXv3tXMK2Sqk5T5QfmyAaANGbxxodvqEdnPPNHLJtwWt3CDIyMtjA/Os6H4i6e3je78PzCVWgUbXWByCeD1AxjB615940tdZ1Ky1qG7tdcm1IOfs0drvFts5weOGOMcfWuthS60zxbcrcWV4y6jpoiimhgLqj46MR93pQB0w8caCdV/s/wC1sJi/lhjEwjLdMb8bf1roK8W0Xw1Y2sdnpOv2viSa9t7pGPlvI1qWVgQ+emMjNe0KNqgDsMUALRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABVGbRrG41u21aWHde2sTwxSbj8qsQWGOn8IoooAvUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQB/9k=)we reduced the number of hidden layers, but with Relu activation and using Euclidian for distance calculation.

### Optimizer and Learning Rate

* When first compiled using SGD with learning rate 0.000001, the rate of reduction in loss was very slow. Then we changed to the Adam optimizer, it actually performed better for our model.
* We have kept the low as it makes the model easy to converge

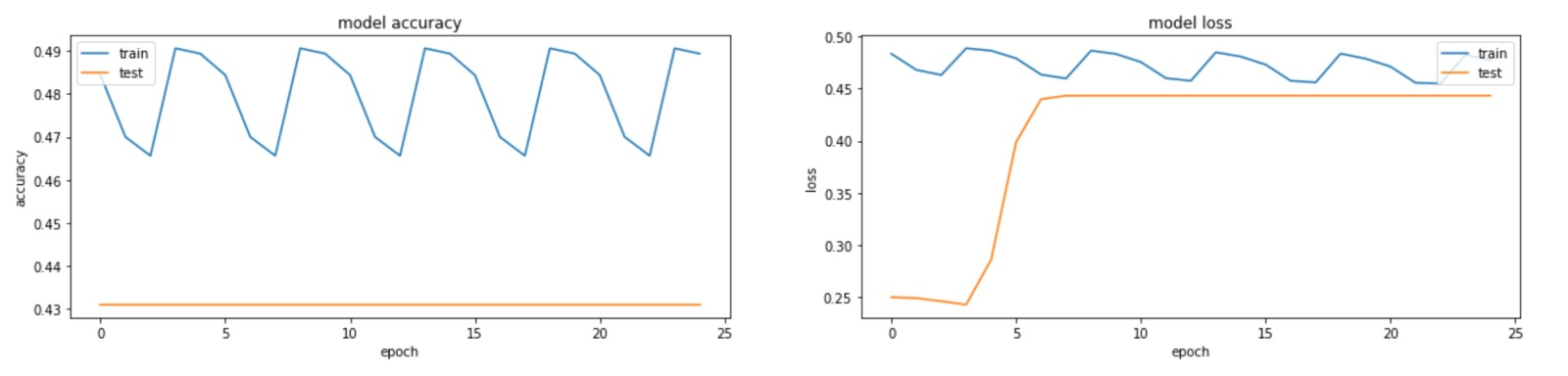
|  |
| --- |
| optimizer = Adam(lr = 0.00006) |

### Loss Function

* We have used Binary CrossEntropy loss function to train our model . The accuracy was high but without any overfitting.

|  |
| --- |
| model.compile(loss="binary\_crossentropy", metrics=['accuracy'], optimizer=optimizer) |

## Using Euclidean Distance with **Contrastive Loss**



|  |
| --- |
| model.compile(loss=contrastive\_loss, optimizer=optimizer, metrics=['accuracy']) |

## Using Euclidean Distance with **Triplet Loss**

The custom Triplet loss function used did not help the model to learn and hence the accuracy didn’t improve much after changing learning rate, embedding size or the model layers. The accuracy was stuck at 0.5. AUC curve yields very much like Figure 1- ROC for Cosine Distance

# Comparison of model performances

The below table represents the accuracy and loss associated with various distance measures and losses used. Weighted L1 and Euclidean using Binary CrossEntropy yielded higher accuracy, least overfitting and minimal loss.

|  |
| --- |
| **Accuracy vs Loss for various Distance Measure** |
| Weighted L1 |
| Cosine |
| Euclidian |
| Euclidian with **Contrastive Loss** |

# Model Deployment

We have deployed two different models in AWS and in local machine using our Flask API server. Using Postman app, we test our recorded voices.

1. Prediction using a same/different classifier model

|  |
| --- |
|  |

1. Prediction using k-nearest neighbour of the embeddings generated from an embedding model

|  |
| --- |
|  |

### API via two routes

1. GET '/' - Gives the API description
2. POST '/predict' - Used to predict speaker  
   - Body params:  
   ----file: accepts audio that needs to be predicted in .wav format  
   ----prediction\_mode: accepts either "using\_knn" or "using\_classifier"

### Next steps in Deployment

1. Create another API to add speakers, their names and audio .wav files to the saved dataset.
2. Deploy API to a webpage to enhance user experience.

# References

Florian Schroff, Dmitry Kalenichenko, James Philbin. “FaceNet: A Unified Embedding for Face Recognition and Clustering”

17 Jun 2015. <https://arxiv.org/pdf/1503.03832.pdf>

Prasad Ramesh. “Face recognition using siamese networks [Tutorial]”  
Packt>. Feb 25, 2019. <https://hub.packtpub.com/face-recognition-using-siamese-networks-tutorial/>

“Docs » Mnist siamese”

Keras Documentation. <https://keras.io/examples/mnist_siamese/>

Raia Hadsell, Sumit Chopra, Yann LeCun. “Dimensionality Reduction by Learning an Invariant Mapping”

Yann LeCun's Publications. Nov 2005. <http://yann.lecun.com/exdb/publis/pdf/hadsell-chopra-lecun-06.pdf>

Oscar Knagg. “Building a Speaker Identification System from Scratch with Deep Learning”  
Analytics Vidhya. Oct 3, 2018. https://medium.com/analytics-vidhya/building-a-speaker-identification-system-from-scratch-with-deep-learning-f4c4aa558a56

Harshall Lamba. “One Shot Learning with Siamese Networks using Keras”

Medium - Towards Data Science. 21 Jan 2019. https://medium.com/@harshall.lamba/one-shot-learning-with-siamese-networks-using-keras-17f34e75bb3d

Prabhnoor Singh. “Siamese Network Keras for Image and Text similarity”

Medium. 21 Jan 2019. https://medium.com/@prabhnoor0212/siamese-network-keras-31a3a8f37d04