Enhancement of Speech Recognition Network Using MLP

Narges Baba Ahmadi, Niloufar Baba Ahmadi

Prof. Hugo Van Hamm

A Preliminary Study Report
Department of Electrical Engineering ESAT
Katholieke Universiteit Leuven

September 13, 2022

Contents

List of Figures

List of Tables

1 Introduction

Given the results of a speech recognition network; We evaluated the accuracy of this network's predicted probabilities and found interesting results. The network was uncertain about its predictions although it shouldn't have been as these low probabilities gave us a high accuracy. In this paper we try to solve this problem by giving the prediction patterns to an MLP model and enhancing the probability evaluation per character.

1.1 Data Description

The data is a sample of standard dutch spoken in Flanders and the Netherlands, also known as CGN, with approximately a selection of one million words. Different dimensions underlying the variation that can be observed in language use were also taken into account which led to distinguish a number of components[1].
Table (1.1) represents the different components.

label	description
a.	Spontaneous conversations ('face-to-face')
b.	Interviews with teachers of Dutch
c.	Spontaneous telephone dialogues (recorded via a switchboard)
d.	Spontaneous telephone dialogues (recorded on MD via a local interface)
e.	Simulated business negotiations
f.	Interviews/discussions/debates (broadcast)
g.	(political) Discussions/debates/meetings (non-broadcast)
h.	Lessons recorded in the classroom
i.	Live (eg sports) commentaries (broadcast)
j.	Newsreports/reportages (broadcast)
k.	News (broadcast)
l.	Commentaries/columns/reviews (broadcast)
m.	Ceremonious speeches/sermons
n.	Lectures/seminars
o.	Read speech

Table 1.1: Components

The researchers who trained an ESPnet on CGN data, only used the Flemish data in training and evaluation. Also components c and d were excluded.

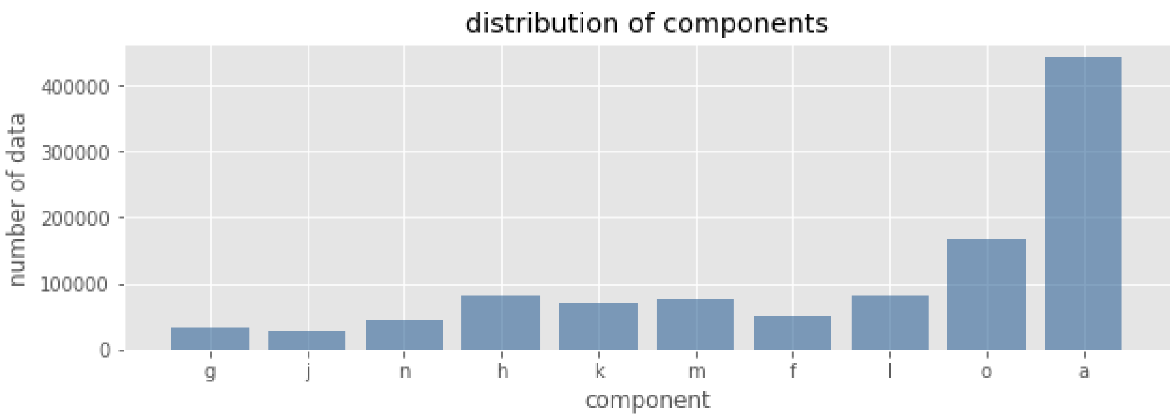


Figure 1.1: Distribution of components

1.2 Problem Statement

Each character in a sentence has a certain probability inserted inside an array with 37 elements and each time the network makes a guess, the character with the highest probability is chosen. By comparing the accuracy of the first-best character in a given interval and the probability given to that character, some insights can be gained and therefore 1.2 is illustrated.

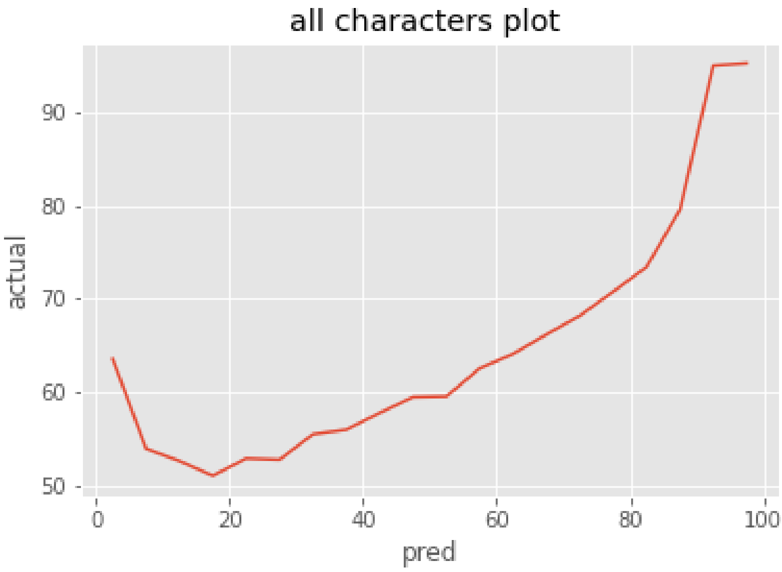


Figure 1.2: Actual probability vs the predicted probability of the predicted character

As is evident in the illustration (1.2), There is a sharp tail between 0.1% to 5% and also the actual probability is always greater than 50% for predicted probabilities as low as 5% and below; hence, the network is 'unsure' of the characters it chooses despite having high accuracy for these low probabilities.

1.3 Objectives

Where does the tail come from?

- Difference in distribution of each 37 characters in the data set could be the issue.
- Having different components could be another reason.
- The difference in data distribution between the intervals is also be a possibility.

How can we enhance the results of this network? In other words, how to design a network with better confidence scores?

- Our hypothesis is that: By using MLP, we can map the actual probabilities to the predicted probabilities.

Which mapping is the best and how to measure that?

- We will try to find the answer to this by applying different mappings and comparing the results.

2 Methods

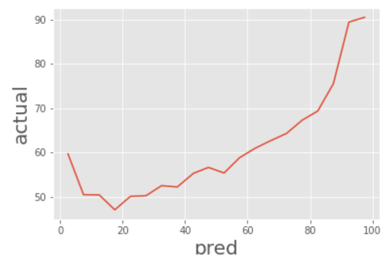
To illustrate plot 1.2, We went through the 2D numpy arrays that we had as the result of the first network. Each element of this 2D array is an array with the length 37 and each of these 37 elements co-responses to a symbol of our character's list. Table 2.1 will convey the elements and their co-responding codes.

Character List			
Characters	Code	Characters	Code
<blank>	0	p	20
X	1	q	21
’	2	r	22
-	3	s	23
<space>	4	t	24
a	5	u	25
b	6	v	26
c	7	w	27
d	8	x	28
e	9	y	29
f	10	z	30
g	11	ç	31
h	12	é	32
i	13	ë	33
j	14	ï	34
k	15	ö	35
l	16	<eos>	36
m	17		
n	18		
o	19		

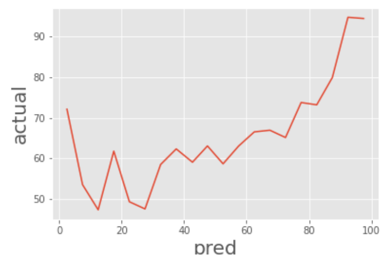
Table 2.1: Character's encoding

After going through the the mentioned numpy array, all of the predicted probabilities were found. Next, a tool was needed to check whether the predicted character was the correct one. We used sclite tool which is a tool for scoring and evaluating the output of speech recognition systems. The program compares the hypothesized text output by the speech recognizer to the correct, or reference text and it is not case sensitive(for more detailed information, visit [2]). The mismatches found by sclite consist of three different types: Insertion, deletion and substitution. In the scope of this project only deletion and substitution were covered. As we don't have any probability for inserted indexes, it is quite difficult to cover this type of mismatch. Using the results of sclite, actual probabilities of characters were calculated. However, two problems stood out after examining the result plot (1.2); firstly, there exists a sharp tail at the beginning of the curve and secondly, there is a huge gap between the predicted and actual probabilities.

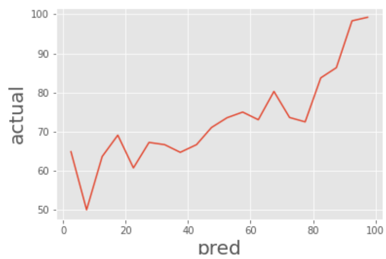
Initially, our focus was on figuring out where the tail comes from. The very first hypothesis was that: considering all different components as one could be causing this problem. Illustrating the same plot for each component would give us some insights.



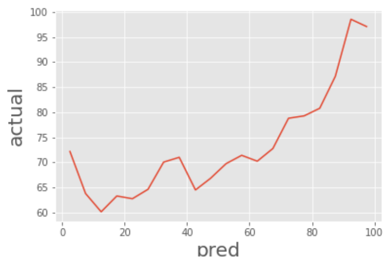
Component a



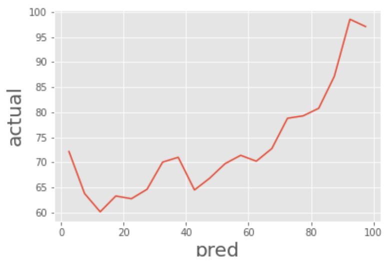
Component f



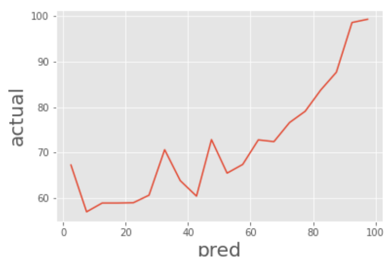
Component g



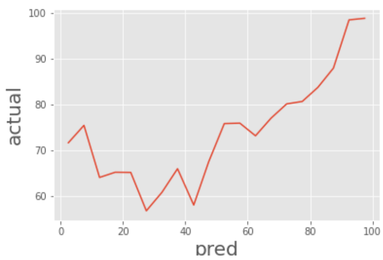
Component h



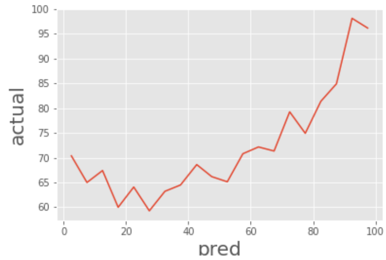
Component j



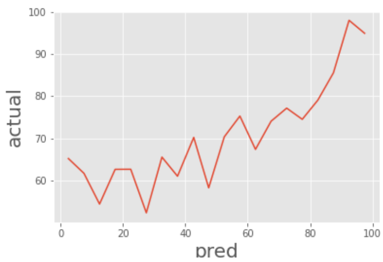
Component k



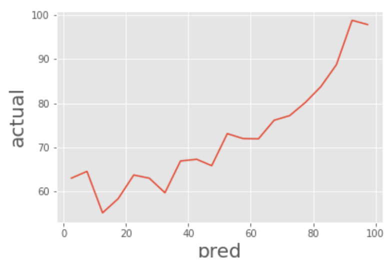
Component l



Component m



Component n

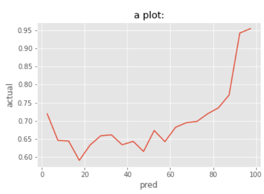


Component o

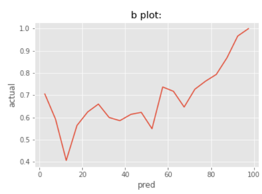
Figure 2.1: Plots for different components

As shown in the figure 2.1, there exists a tail for all of these components which proves our hypothesis to be wrong.

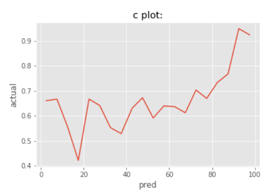
The second assumption was that there might be some characters which are always predicted wrong or right and they might be the reason for the tail. So the decision was to draw such plot for each of the 37 characters.



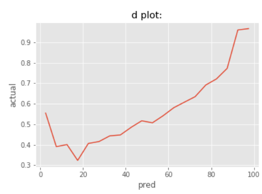
Character a



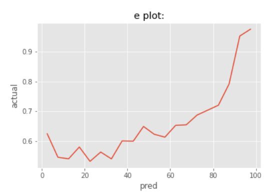
Character b



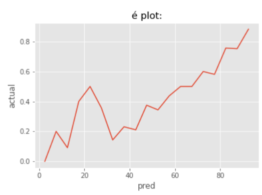
Character c



Character d



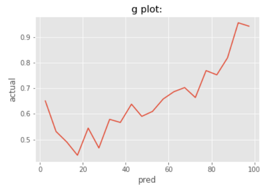
Character e



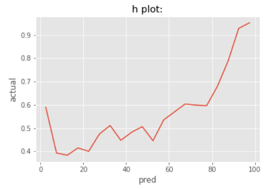
Character é

Character ë

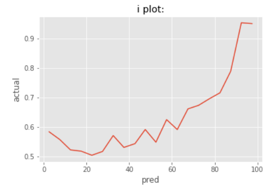
Character f



Character g

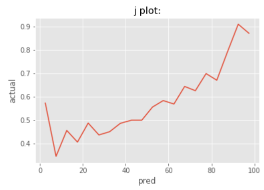


Character h

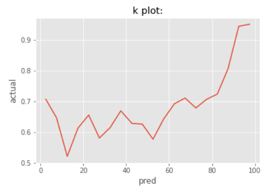


Character i

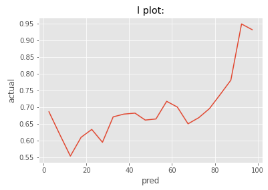
Character ï



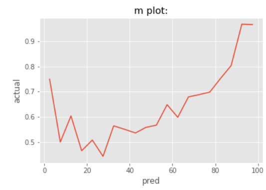
Character j



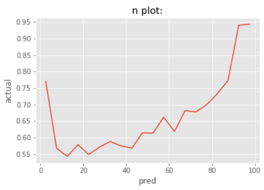
Character k



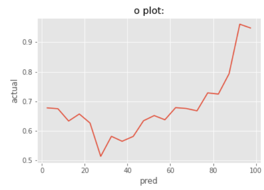
Character l



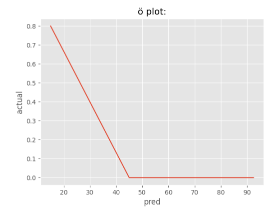
Character m



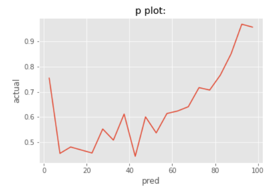
Character n



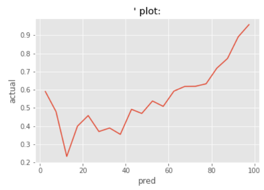
Character o



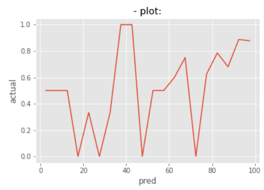
Character ö



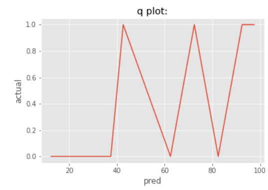
Character p



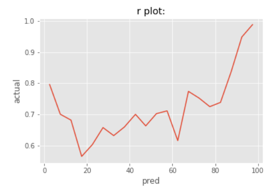
Character ’



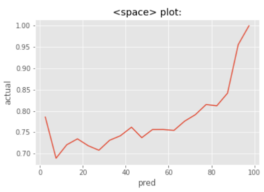
Character -



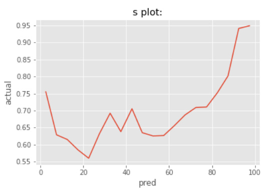
Character q



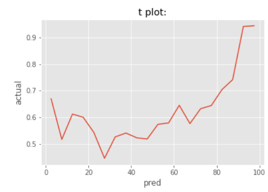
Character r



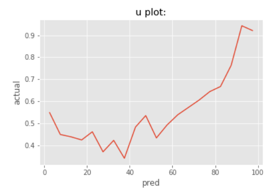
Character space



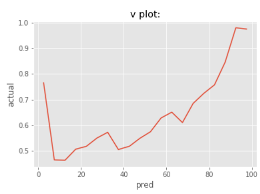
Character s



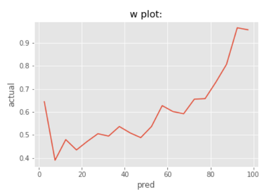
Character t



Character u



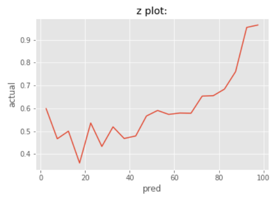
Character v



Character w

Character x

Character y



Character z

Figure 2.2: Plots for different characters

Some of the plots have a strikingly bizarre shape; However, taking figure 2.3 in account, low distribution of those characters is the reason for this behavior.

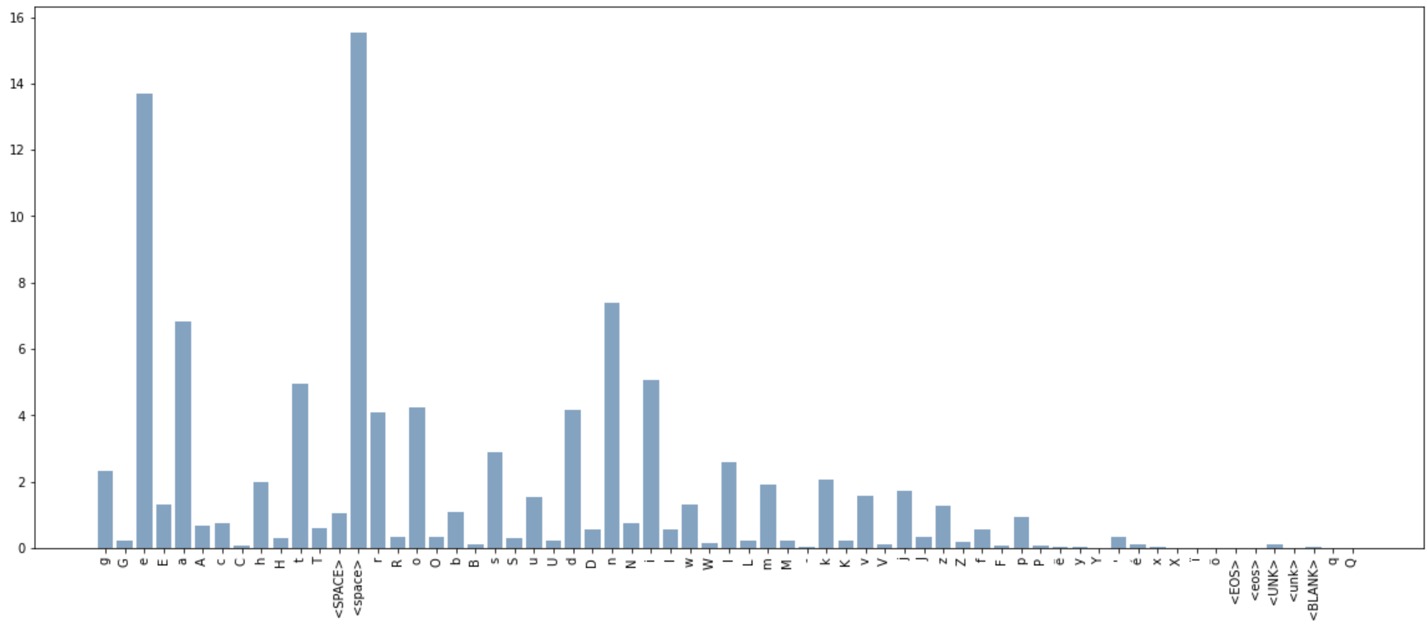


Figure 2.3: Distribution of the characters

Table 2.2 shows the number of data per character.

Character List			
Characters	Number of data	Characters	Number of data
'	6104	r	48555
-	578	s	33632
<space>	182990	t	60744
a	81837	u	19542
b	13201	v	18199
c	9726	w	16301
d	50341	x	2872
e	164359	y	649
f	7071	z	15289
g	28697	ç	1
h	26739	é	1608
i	61103	ë	405
j	22062	ï	47
k	25042	ö	28
l	31718		
m	23195		
n	88902		
o	49802		
p	10995		
q	68		

Table 2.2: Character's frequency

Since almost all plots in figure 2.2 has a tail, the hypothesis is once more rejected which brings us to the last hypothesis that says this tail is caused due to the difference between the amount of data in each interval. Since only a few data points exist in those intervals, the results are not generalized and simply are errors of the network. Figure 2.4 illustrates this occurrence.

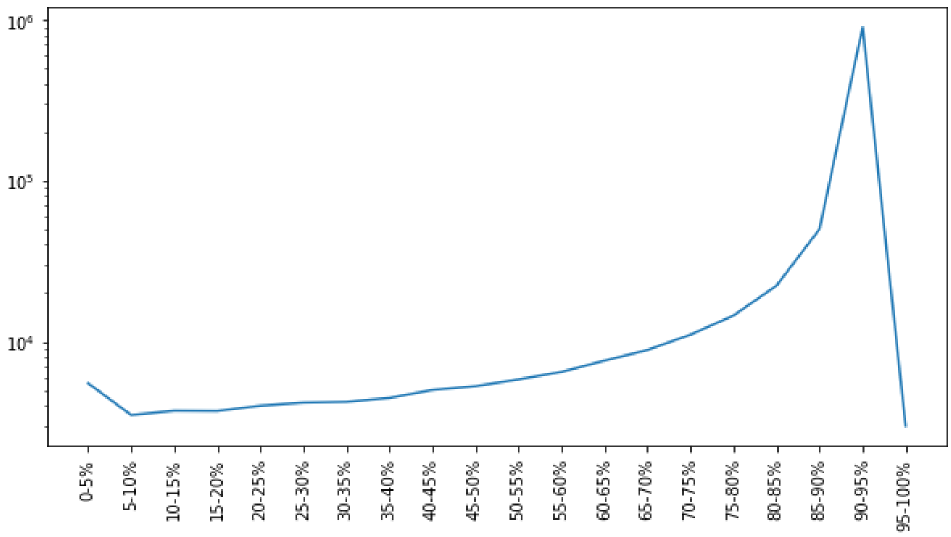


Figure 2.4: Distributions among different intervals

By combining the first intervals, the tail problem got solved as evident in figure 2.5 , 2.6 and 2.7. This solves the problem because the mean of the points were calculated and illustrated on an bigger interval. It is worthwhile mentioning that only 4% of data exists before the 55% interval. Using table 2.3, it is evident that the predicted character is correct with at least 55% accuracy and therefore any data point under 55% is an error of the network.

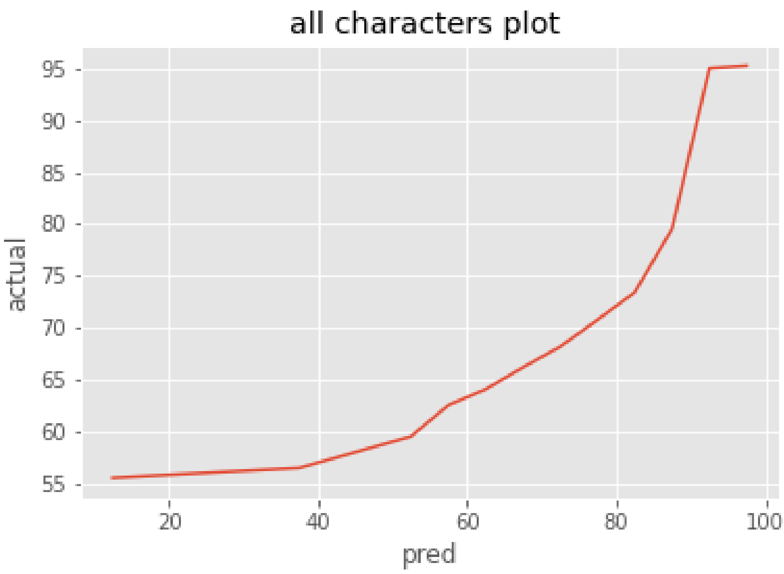
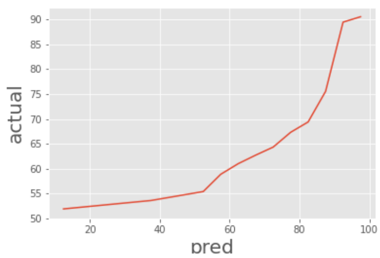
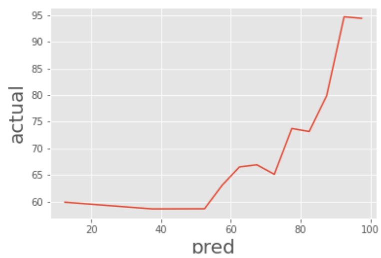


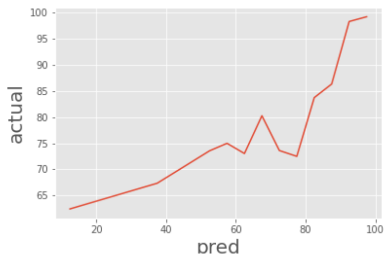
Figure 2.5: Actual probability vs the predicted probability of the predicted character(after combination)



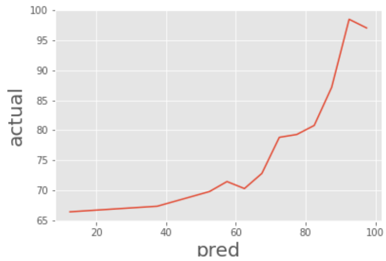
Component a



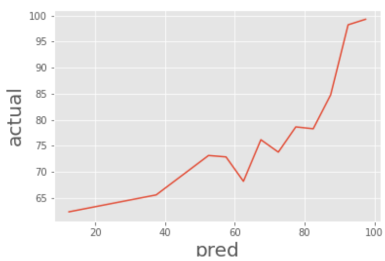
Component f



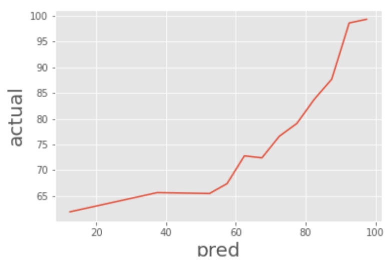
Component g



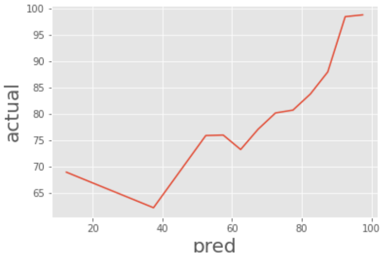
Component h



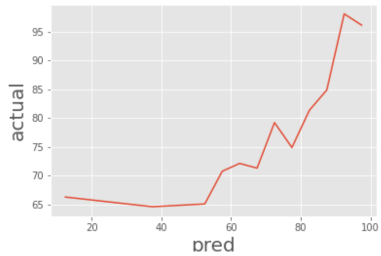
Component j



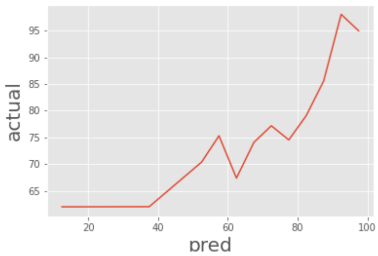
Component k



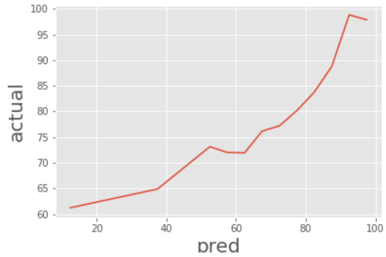
Component l



Component m

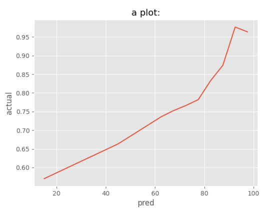


Component n

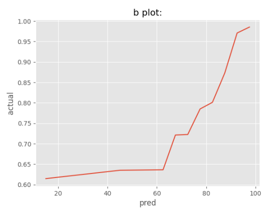


Component o

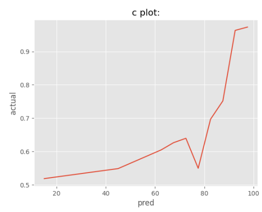
Figure 2.6: Plots for different components (after combination)



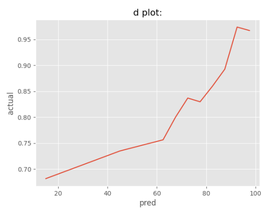
Character a



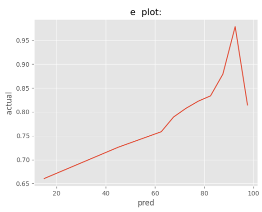
Character b



Character c



Character d

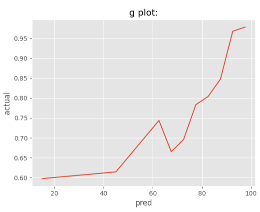


Character e

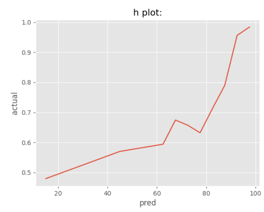
Character é

Character ë

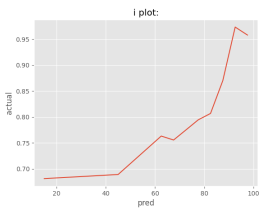
Character f



Character g

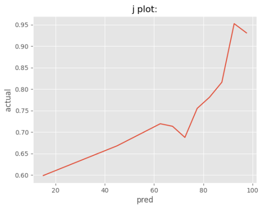


Character h

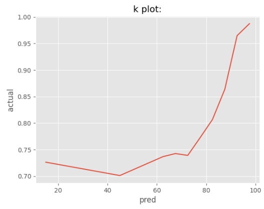


Character i

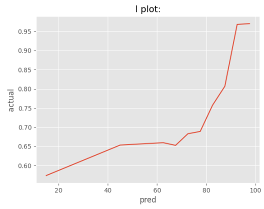
Character ï



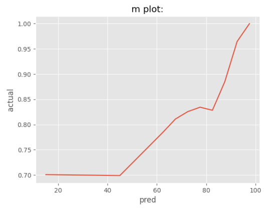
Character j



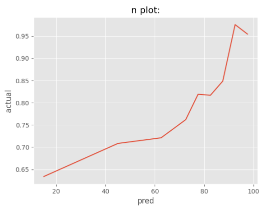
Character k



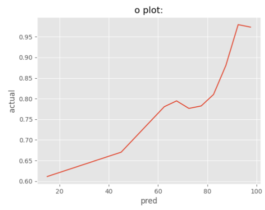
Character l



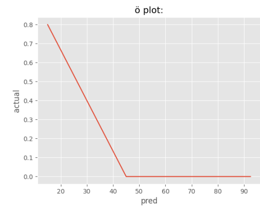
Character m



Character n

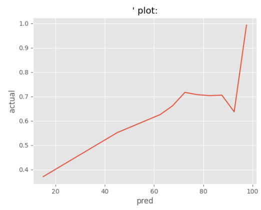


Character o

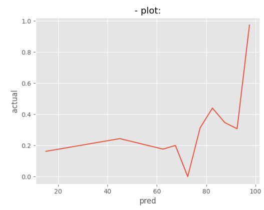


Character ö

Character p

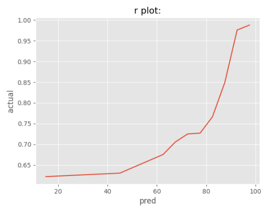


Character ′

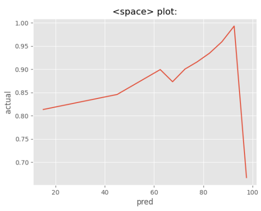


Character -

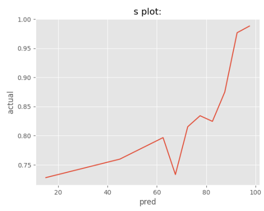
Character q



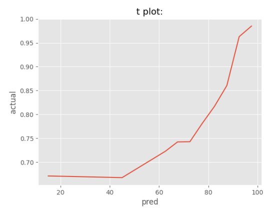
Character r



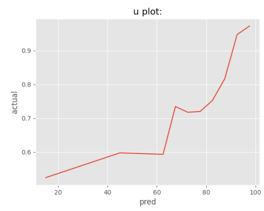
Character space



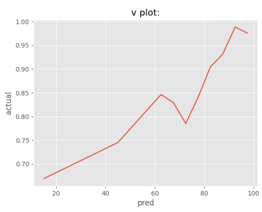
Character s



Character t

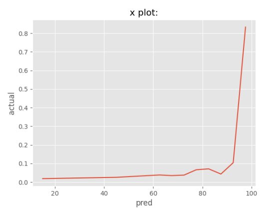


Character u

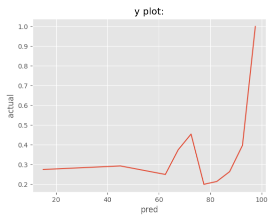


Character v

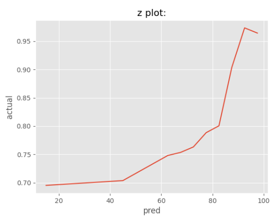
Character w



Character x



Character y



Character z

Figure 2.7: Plots for different characters (after combination)

2.1 Dataset Construction

To enhance the results of the original network, we decided to train an MLP network but first the network's features and targets needed to be constructed. The features were extracted from the predicted probabilities; to put it in another way, the outputs of the previous network are the inputs of the new one. To create the targets of the network, the whole CGN data set was examined and look-up table 2.3 was formed.

	0-5%	5-10%	10-15%	15-20%	20-25%	25-30%	30-35%	35-40%	40-45%	45-50%
First best	0.63562	0.53960	0.52623	0.51052	0.52902	0.52753	0.55518	0.55991	0.57848	0.59485
Second best	0.00035	0.00169	0.00267	0.00240	0.00175	0.00272	0.00304	0.00209	0.00376	['N']
Third best	0.00035	0.00210	0.00237	0.00263	0.00314	['N']	['N']	['N']	['N']	['N']
4th best	0.00032	0.00153	0.00240	['N']	['N']	['N']	['N']	['N']	['N']	['N']
5th best	0.00028	0.00089	0.00719	['N']	['N']	['N']	['N']	['N']	['N']	['N']
6th best	0.00022	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']
7th best	0.00019	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']
8th best	0.00018	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']
9th best	0.00015	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']
10th line	0.00012	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']
11th best	0.0.00010	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']
12th best	0.00010	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']
13th best	8.7861e-05	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']
14th best	7.1971e-05	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']
15th best	7.7579e-05	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']
16th best	5.7016e-05	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']
17th best	6.0754e-05	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']
18th best	5.3277e-05	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']
19th best	4.3930e-05	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']
20th best	6.0754e-05	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']
21th best	5.7950e-05	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']
22th best	6.1689e-05	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']
23th best	5.7016e-05	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']
24th best	5.8885e-05	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']
25th best	6.9167e-05	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']
26th best	6.6363e-05	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']
27th best	7.7579e-05	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']
28th best	6.4493e-05	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']
29th best	6.9167e-05	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']
30th best	7.2905e-05	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']
31th best	7.2905e-05	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']
32th best	6.5428e-05	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']
33th best	7.2905e-05	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']
34th best	0.0001	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']
35th best	9.1599e-05	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']
36th best	9.9077e-05	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']
37th best	0.0001	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']

	50-55%	55-60%	60-65%	65-70%	70-75%	75-80%	80-85%	85-90%	90-95%	95-100%
First best	0.59556	0.62573	0.64099	0.66208	0.68199	0.70791	0.73446	0.79555	0.95009	0.95252
Second best	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']	['N']

* The header row represents the predicted probabilities.
* The cells represent the actual probability.
* In the context of this table, ['N'] means null.
* The rest of the rows are all null after 50%.

Table 2.3: Look-up table

For each index of each record, first we found its rank among the 37 elements of the record and after finding the corresponding interval of that number and using table 2.3, we replaced the value of that index with the correct value.

2.2 Models

2.2.1 Cross-entropy loss

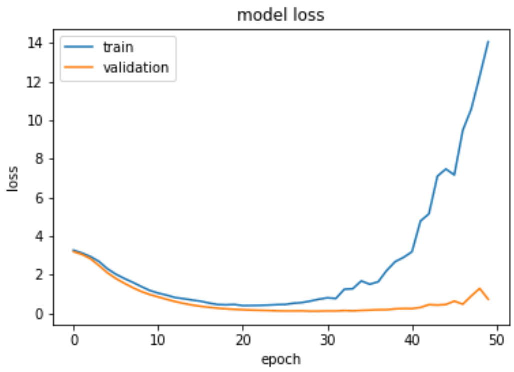
The first model we trained was a sequential model made of four layers. Figure 2.8 illustrates the model. The first three layers had Relu activation function and the last layer had Softmax as the activation function. Categorical cross-entropy was used as the loss function and the optimizer was set to Adam.

```
Model: "sequential_16"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_107 (Dense)           (None, 120)               4560

 dropout_42 (Dropout)        (None, 120)               0

 dense_108 (Dense)           (None, 60)                7260

 dropout_43 (Dropout)        (None, 60)                0

 dense_109 (Dense)           (None, 30)                1830

 dropout_44 (Dropout)        (None, 30)                0

 dense_110 (Dense)           (None, 37)                1147

=================================================================
Total params: 14,797
Trainable params: 14,797
Non-trainable params: 0
_________________________________________________________________
```

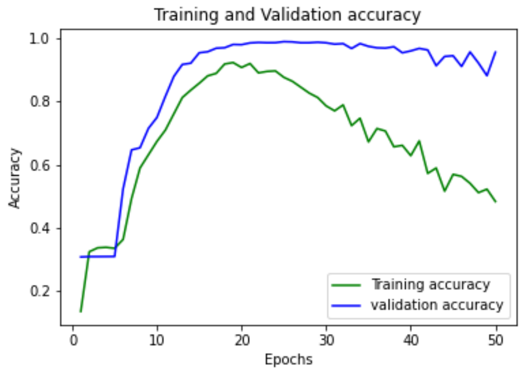
Figure 2.8: Model description (Cross-entropy loss function)

The number of epochs was set to 20 because according to figure 2.9 that gives us the most ideal results.

The accuracy of this model is 98.58% on the train set and 98.61% on the validation set. Figure 2.9 shows the loss curve on the train and validation set.



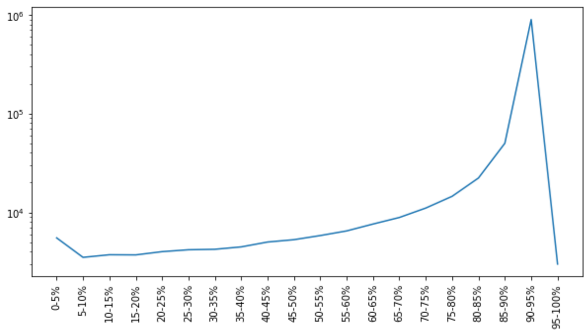
Model's Loss



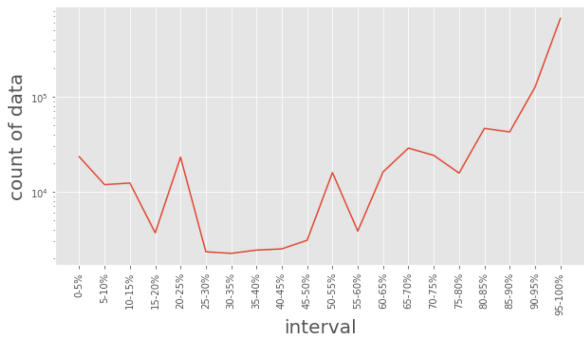
Model's accuracy

Figure 2.9: Loss and accuracy of the model with cross-entropy loss function

The accuracy is 98.54% on the test set with 0.2302 loss. The distribution of first best element in predicted vectors is shown below (2.10).



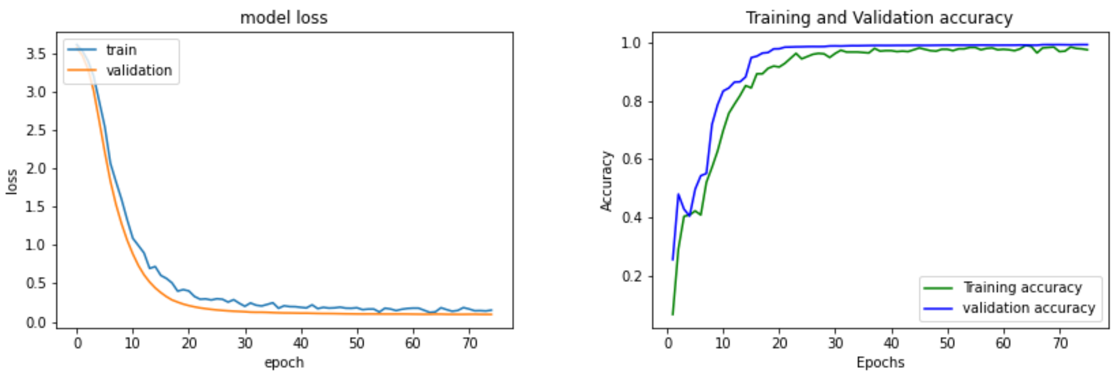
(a) Distribution before training



(b) Distribution after training

Figure 2.10: Comparison between the distribution of predicted character in the model with Cross-entropy loss

The same model was trained on normalized data. According to figure 2.11, 40 epochs are the ideal choice. This model gave us 99.15% accuracy on the train set and 99.16% on the validation set. Figure 2.11 shows the loss curve on the train and validation set.

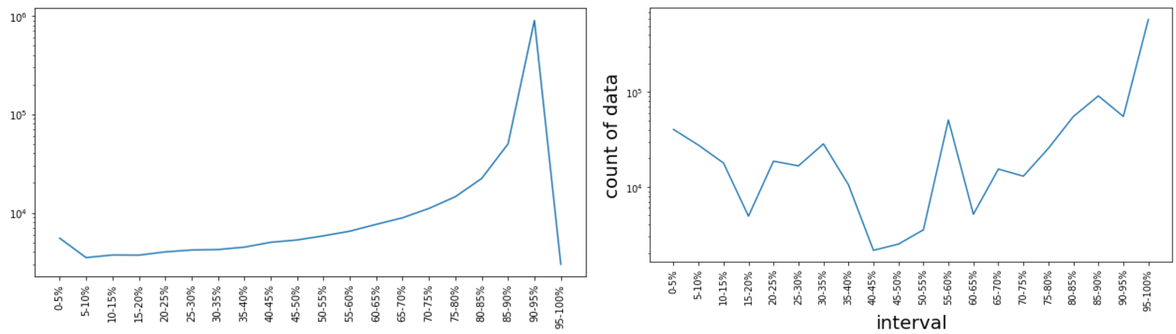


Model's Loss

Model's accuracy

Figure 2.11: Loss and accuracy of the model with cross-entropy loss function trained on nomalized data

The accuracy is 99.14% on the test set with 0.1135 loss. The distribution of first best element in predicted vectors is shown below (2.12).



(a) Distribution before training

(b) Distribution after training

Figure 2.12: Comparison between the distribution of predicted character the model with Cross-entropy loss trained on normalized data

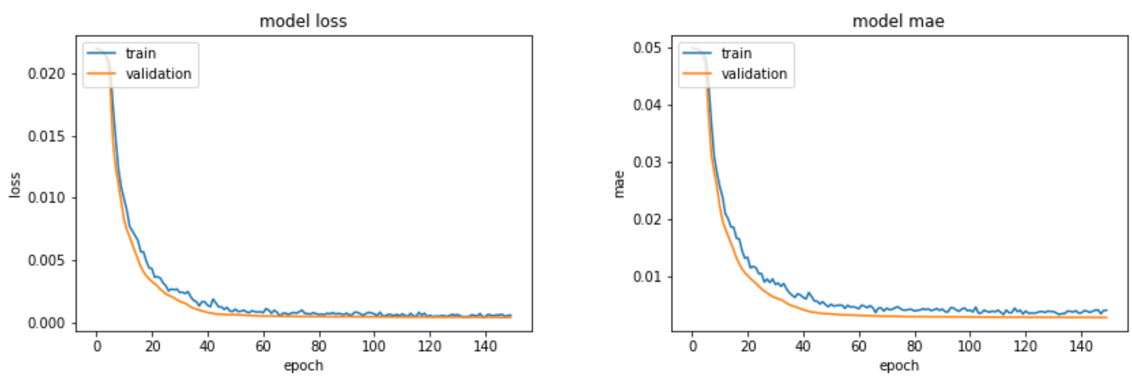
2.2.2 Mean Square Error loss

The second model we trained was a regression model with MSE as the loss function. Figure 2.13 illustrates the model.

```
defining model
Model: "sequential_25"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_143 (Dense)           (None, 120)               4560

 dropout_69 (Dropout)        (None, 120)               0

 dense_144 (Dense)           (None, 60)                7260

 dropout_70 (Dropout)        (None, 60)                0

 dense_145 (Dense)           (None, 30)                1830

 dropout_71 (Dropout)        (None, 30)                0

 dense_146 (Dense)           (None, 37)                1147

=================================================================
Total params: 14,797
Trainable params: 14,797
Non-trainable params: 0
```

Figure 2.13: Regression model description

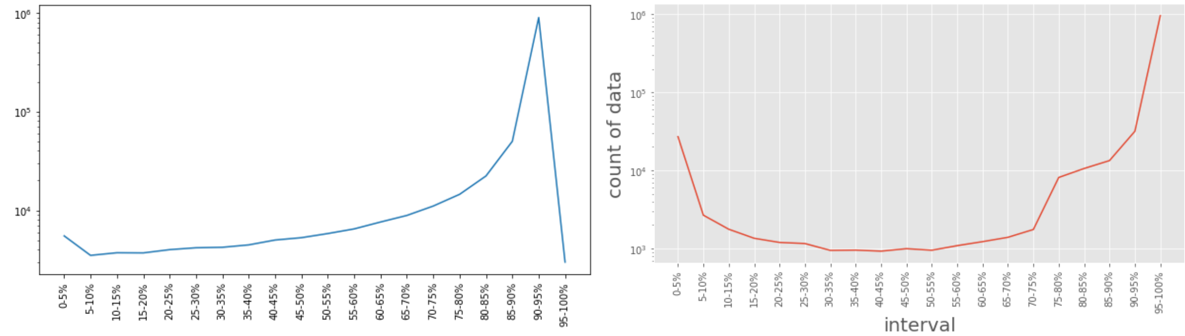
The first three layers had Relu activation function and the last layer had Softmax as the activation function. The optimizer was set to Adam and MAE was used to evaluate the results. Figure 2.14 shows the loss curve on the train and validation set which shows that the optimum epoch is 60. The MAE equals 0.0032 on the train set and 0.0031 on the validation set.



Model's MSE Model's MAE

Figure 2.14: MSE and MAE of the regression model

The MAE equals 0.0032 on the test set with 5.5738e-04 loss. The distribution of first best element in predicted vectors is shown below (2.15).



(a) Distribution before training

(b) Distribution after training

Figure 2.15: Comparison between the distribution of predicted character in the regression model

It might seem like the network is doing a good job, but according to figure 2.16 it is not strictly ascendant anymore.

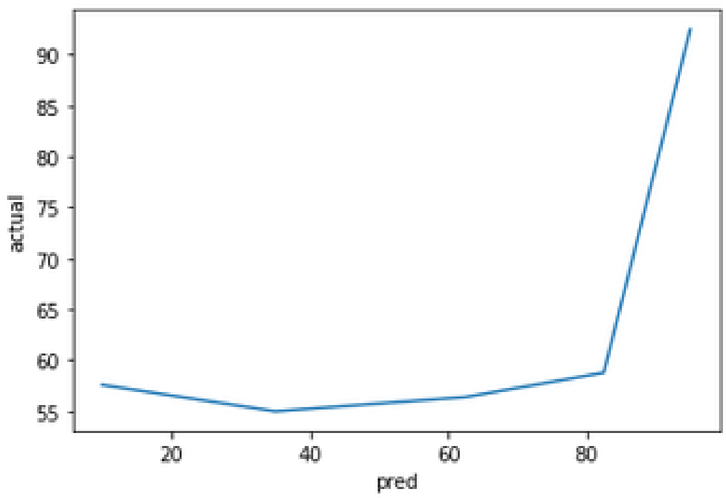
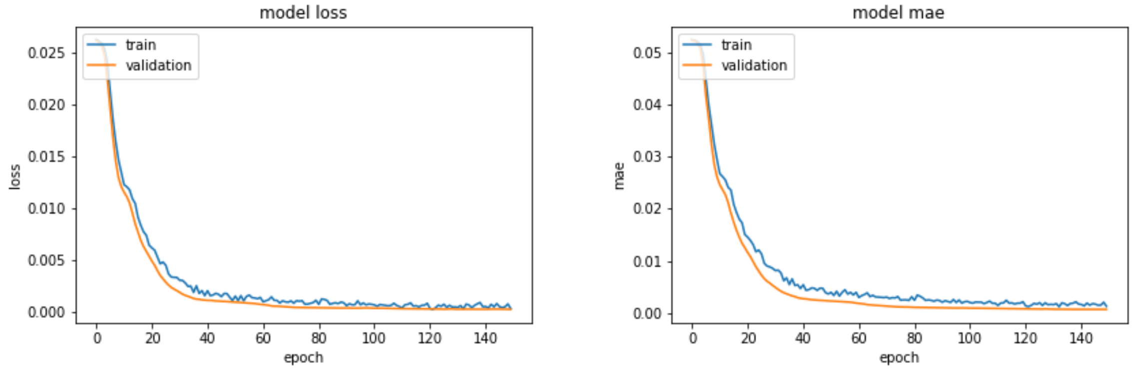


Figure 2.16: Actual probability vs the predicted probability (predicted character)

Now, we run the same model on Normalized data. Figure 2.17 shows the loss curve on the train and validation set. The perfect epoch is once again 60. The MAE equals 0.0021 on the train set and 0.0020 on the validation set.

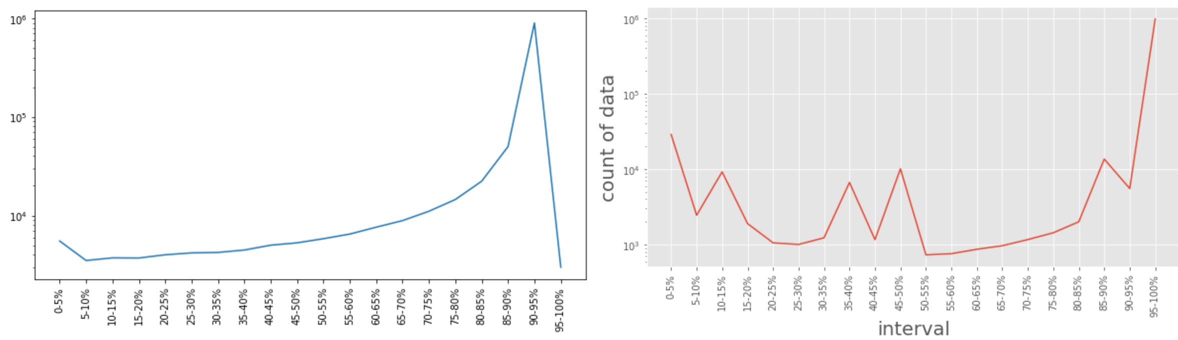
The MAE equals 0.0021 on the test set with 8.3499e-04 loss. The distribution of first best element in predicted vectors is shown below (2.18).



Model's MSE

Model's MAE

Figure 2.17: MSE and MAE of the regression model on normalized data



(a) Distribution before training

(b) Distribution after training

Figure 2.18: Comparison between the distribution of predicted character in the regression model (Normalised data)

2.2.3 Kullback–Leibler divergence

The third model we trained was a sequential model with KLD as its loss function. Figure 2.19 illustrates the model.

Figure 2.20 shows the loss curve on the train and validation set. Accordingly, the optimum number of epochs is 40.

```
Model: "sequential_13"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_45 (Dense)            (None, 120)               4560

 dropout_33 (Dropout)        (None, 120)               0

 dense_46 (Dense)            (None, 60)                7260

 dropout_34 (Dropout)        (None, 60)                0

 dense_47 (Dense)            (None, 30)                1830

 dropout_35 (Dropout)        (None, 30)                0

 dense_48 (Dense)            (None, 37)                1147

=================================================================
Total params: 14,797
Trainable params: 14,797
Non-trainable params: 0
_________________________________________________________________
```

Figure 2.19: Model with KLD loss function

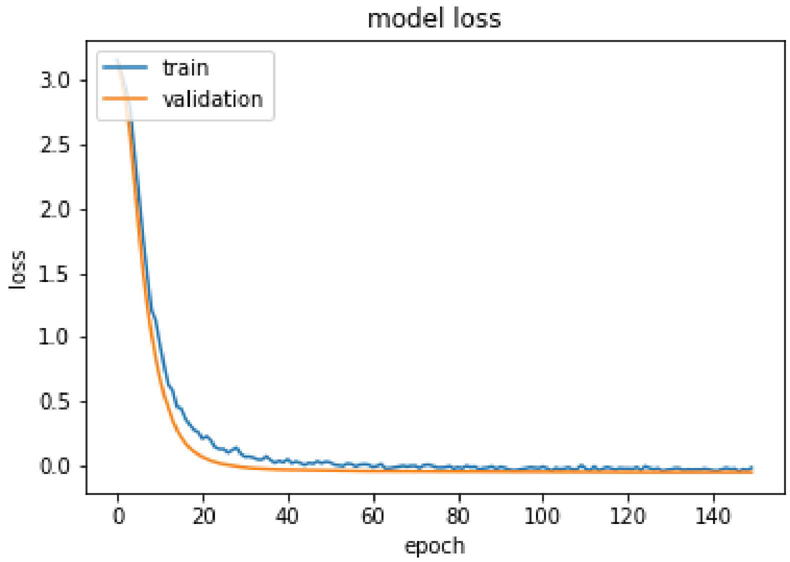
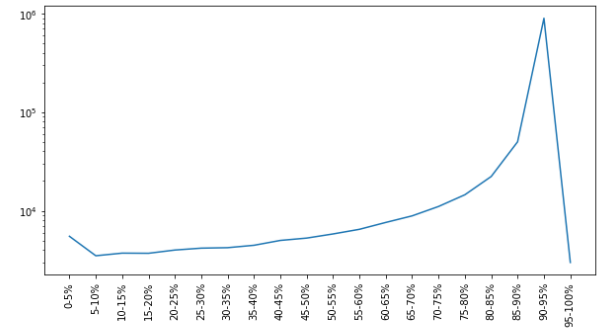
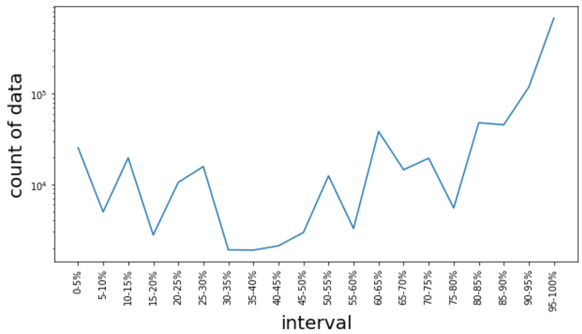


Figure 2.20: Loss of the model with KLD loss function

The loss on the test set is equal to -0.0259. The distribution of first best element in predicted vectors is shown below (2.21).



(a) Distribution before training



(b) Distribution after training

Figure 2.21: Comparison between the distribution of predicted character in the model with KLD loss function

2.3 Data Augmentation

As you can see in the models, the upper intervals are getting better results and lower ones are getting worse. We assumed that low distribution of lower intervals is causing this problem so we decided to add some data to the lower intervals. We increased the number of data in lower intervals by duplicating the previous data points in each interval that had a low distribution. Figure 2.42 shows the distribution after data augmentation.

After changing the data set, we trained the same models on the new data set.

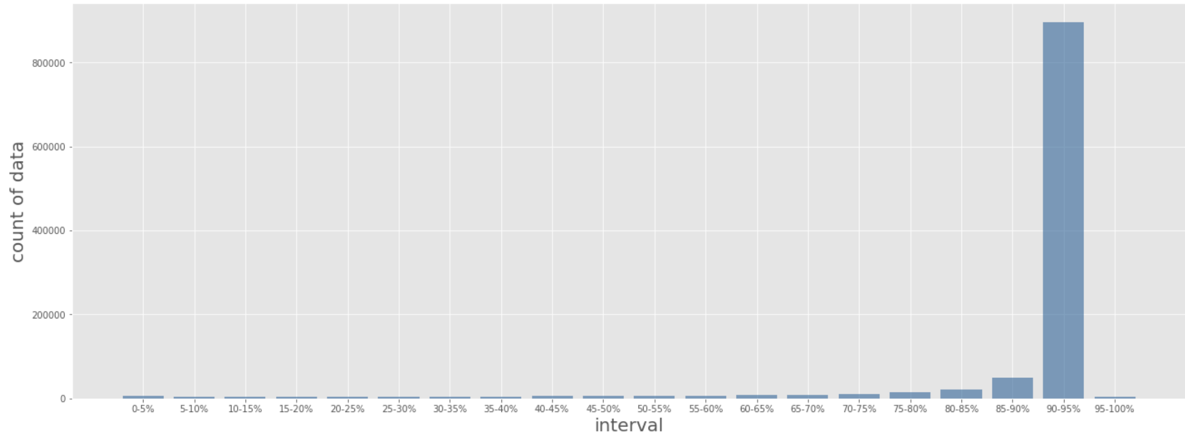
2.3.1 Cross-entropy loss

The models are constructed like before. Figure 2.23 shows the loss curve on the train and validation set. Accordingly, the optimum number of epochs is 60.

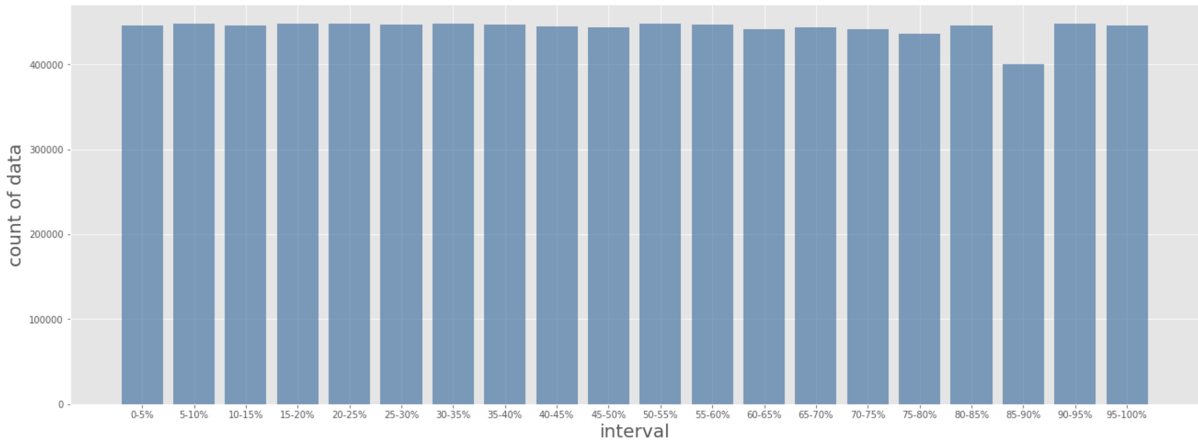
The loss on the test set is equal to 0.2716. The distribution of first best element in predicted vectors is shown below (2.24). As evident from figure 2.24, 3.22% of data exist under 55% threshold.

Figure 2.25 shows the predicted versus actual probability of this network.

As you see, although this model has decreased the amount of data under 55% threshold; it has destroyed the credibility of other intervals and the curve is not



(a) Distribution before data augmentation



(b) Distribution after data augmentation

Figure 2.22: Distribution comparison before and after data augmentation

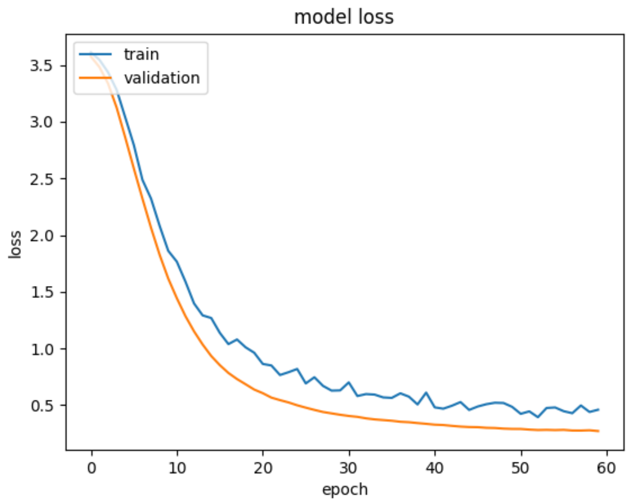
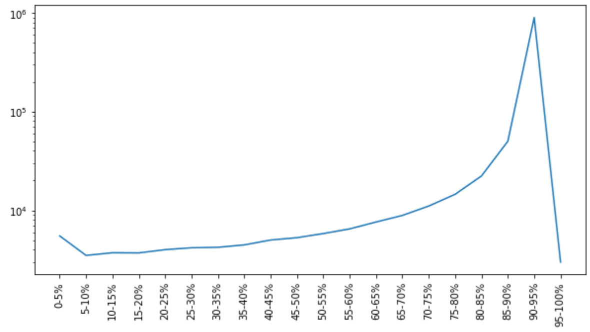
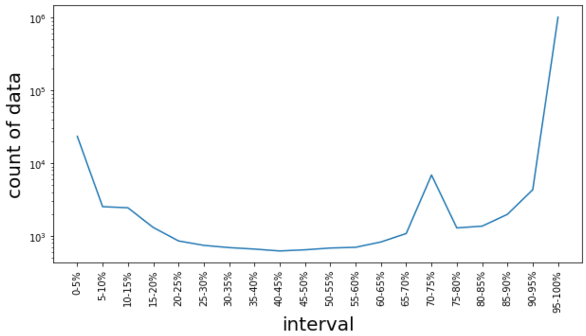


Figure 2.23: Loss of the model with Cross-entropy loss function

ascendant anymore.



(a) Distribution before training



(b) Distribution after training

Figure 2.24: Comparison between the distribution of predicted character in the model with KLD loss function

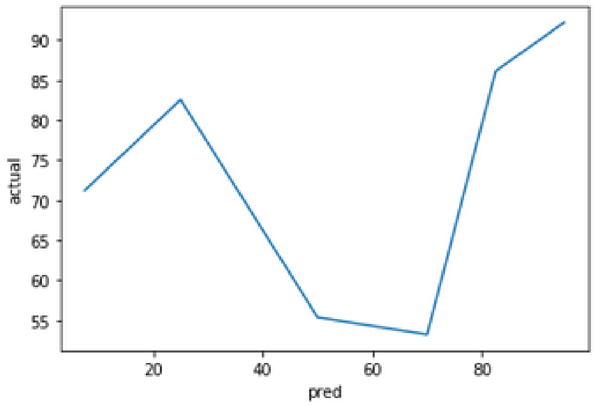


Figure 2.25: Actual probability vs the predicted probability of cross-entropy network (predicted character)

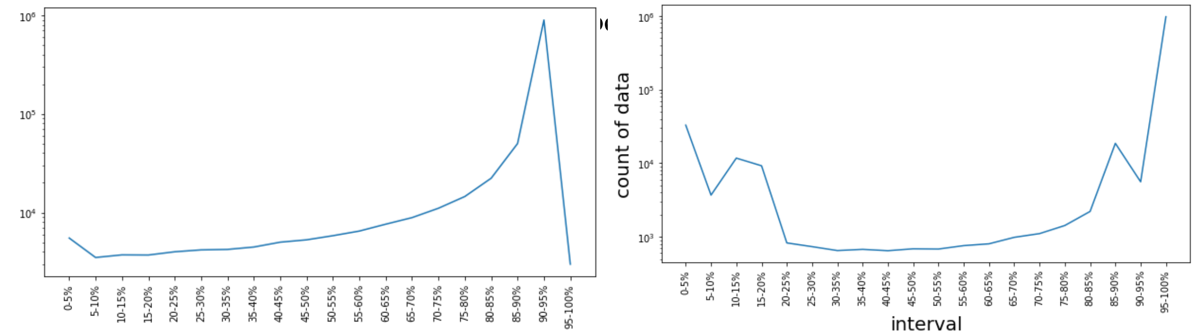
2.3.2 Mean Square Error loss

Figure 2.26 shows the loss curve on the train and validation set. The loss equals 0.0028 and MAE equals 0.00729 on the train set and on the validation set, the loss is equal to 0.0028 and MAE is equal to 0.0073. Accordingly, the optimum number of epochs is 60.

The loss on the test set is equal to 0.0028 with MAE of 0.0073. The distribution of first best element in predicted vectors is shown below (2.27). As evident from figure 2.27, 5.83% of data exist under 55% threshold.

Figure 2.28 shows the predicted versus actual probability of this network.

As you see, not only has this model increased the amount of data under 55%



(a) Distribution before training (b) Distribution after training

Figure 2.27: Comparison between the distribution of predicted character in the model with MSE loss function

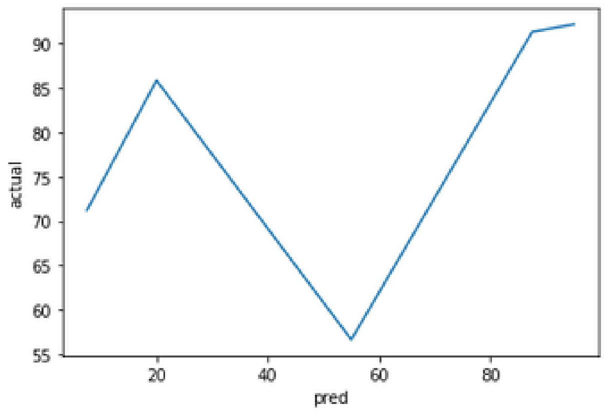


Figure 2.28: Actual probability vs the predicted probability of regression network (predicted character)

threshold, but it has also destroyed the credibility of other intervals and the curve is not ascendant anymore.

2.3.3 Kullback–Leibler divergence

Figure 2.29 shows the loss curve on the train and validation set. The loss equals 0.2119 on the train set and 0.2120 on the validation set. Accordingly, the optimum number of epochs is 50.

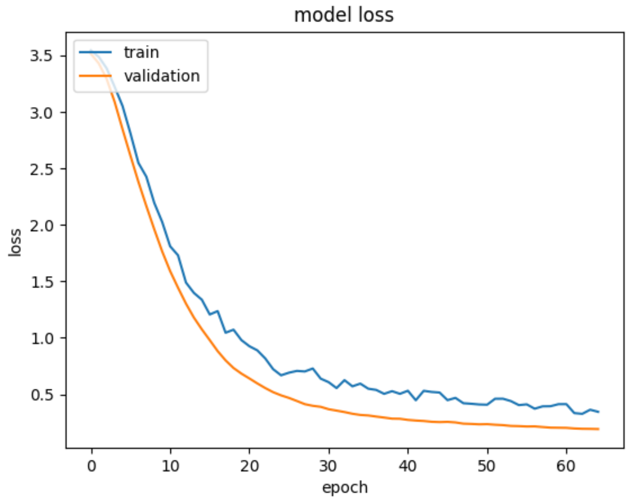
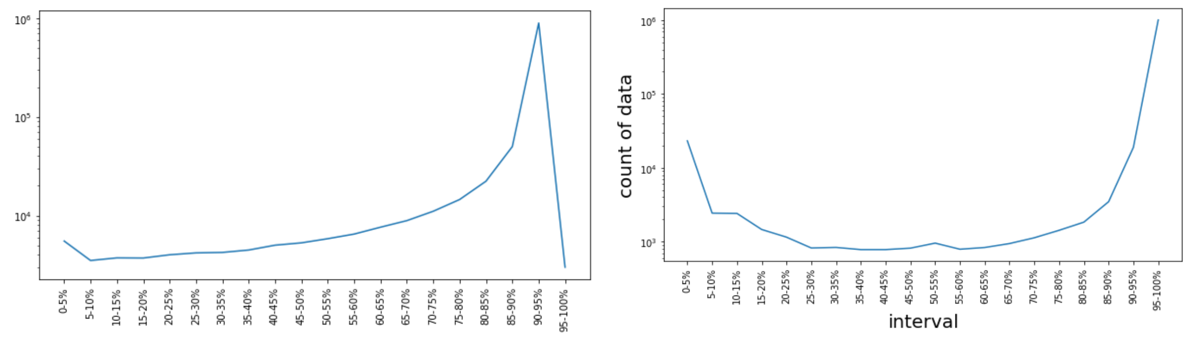


Figure 2.29: Loss of the model with KLD loss function

The loss on the test set is equal to 0.2120. The distribution of first best element in predicted vectors is shown below (2.30). As evident from figure 2.30, only 3.33% of data exist under 55% threshold.



(a) Distribution before training (b) Distribution after training

Figure 2.30: Comparison between the distribution of predicted character in the model with KLD loss function

As you see, although this model has decreased the amount of data under 55% threshold; it has destroyed the credibility of other intervals and the curve is not ascendant anymore.

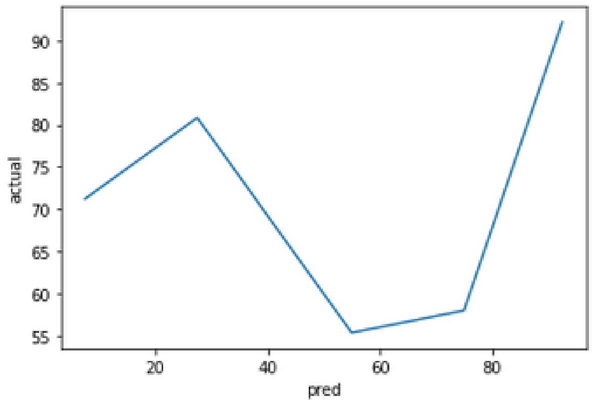
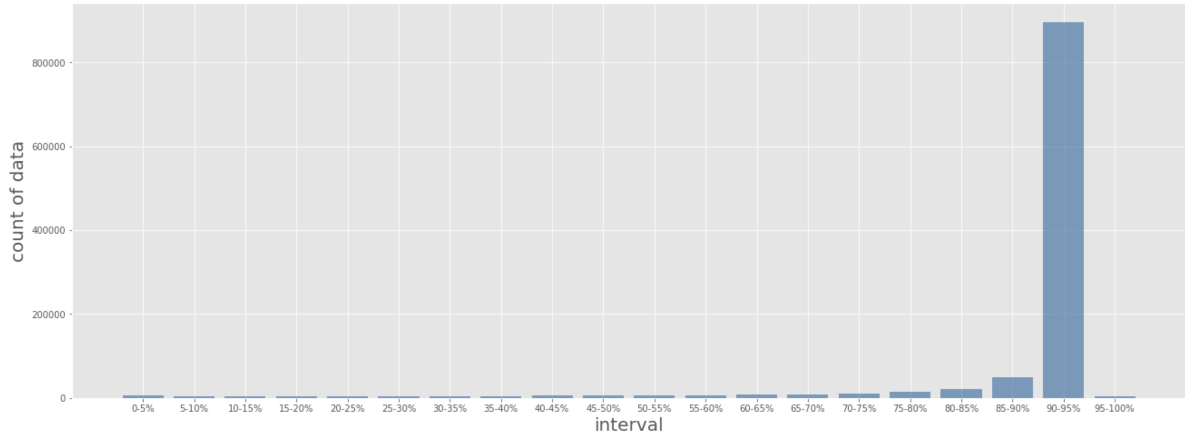


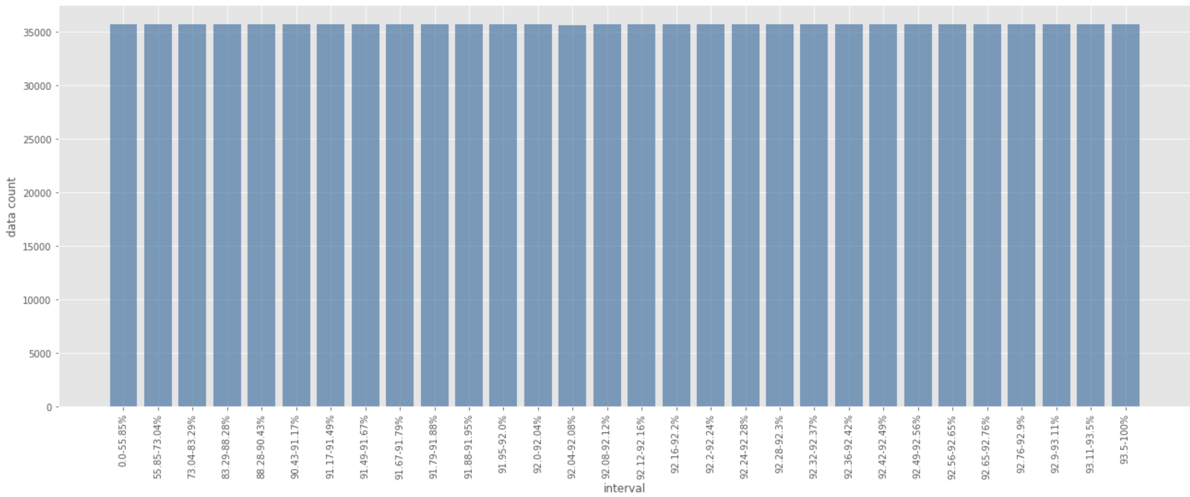
Figure 2.31: Actual probability vs the predicted probability of KLD network (predicted character)

2.4 Dataset Construction (Second approach)

Instead of copying the data to have intervals with equal lengths, we tried to divide intervals based on data distribution. You can find the intervals in appendix A1. The right probability assigned to each interval can also be found in appendix A2.



(a) Distribution before reconstruction



(b) Distribution after reconstruction

Figure 2.32: Distribution comparison before and after reconstruction

2.4.1 Cross-entropy loss

The layers are constructed as before. Figure 2.33 shows the loss curve on the train and validation set. The loss equals 0.1370 on the train set and 0.1363 on the validation set. Accordingly, the optimum number of epochs is 30.

The loss on the test set is equal to 0.1362. The distribution of first best element in predicted vectors is shown below (2.34). As evident from figure 2.34, 5.27% of data exist under 55% threshold.

As you see, not only has this model increased the amount of data under 55% threshold, but it has also destroyed the credibility of other intervals and the curve is

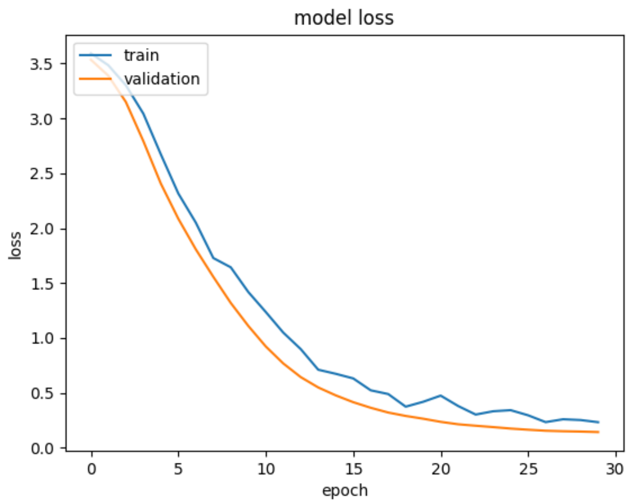
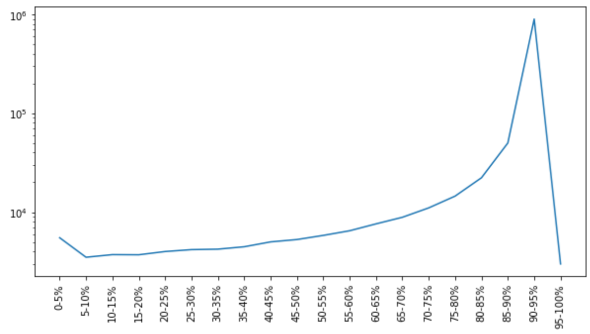
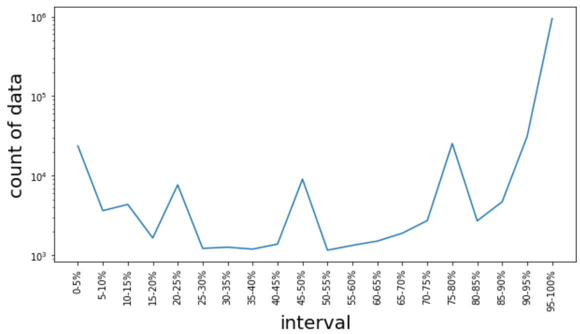


Figure 2.33: Loss of the model with Cross-entropy loss function



(a) Distribution before training



(b) Distribution after training

Figure 2.34: Comparison between the distribution of predicted character in the model with Cross-entropy lo

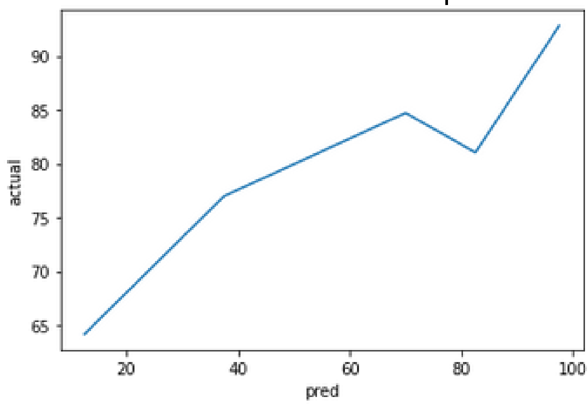


Figure 2.35: Actual probability vs the predicted probability of Cross-entropy network (prediced character)

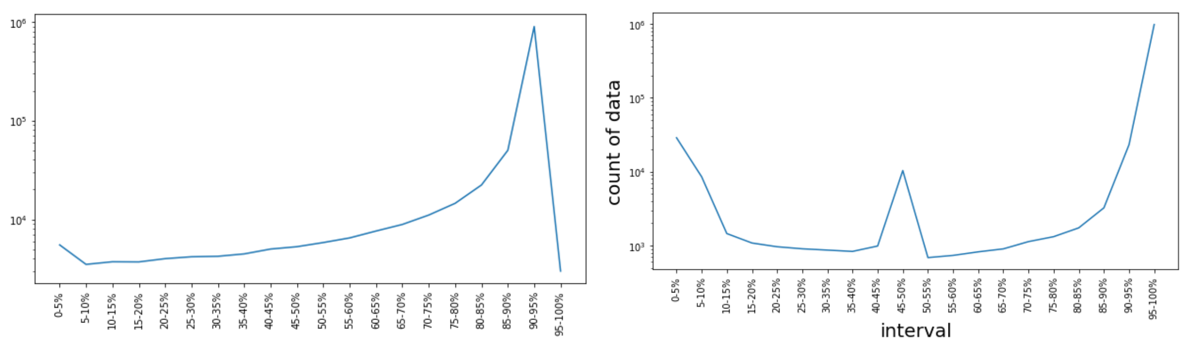
not ascendant anymore.

2.4.2 Mean Square Error loss

The layers are constructed as before. Figure 2.36 shows the loss curve on the train and validation set. The loss equals 4.7747e-04 and MAE equal 0.0014 on the train set and we have the loss equal to 4.7634e-04 and MAE equal to 0.0014 on the validation set.Accordingly, the optimum number of epochs is 60.

Figure 2.36: Loss of the model with MSE loss function

The loss is 4.9205e-04 and the MAE is 0.0014 on the test set. The distribution of first best element in predicted vectors is shown below (2.37). As evident from figure 2.37, 5.17% of data exist under 55% threshold.



(a) Distribution before training

(b) Distribution after training

Figure 2.37: Comparison between the distribution of predicted character in the model with MSE loss function

Figure 2.38 shows the predicted versus actual probability of this network.

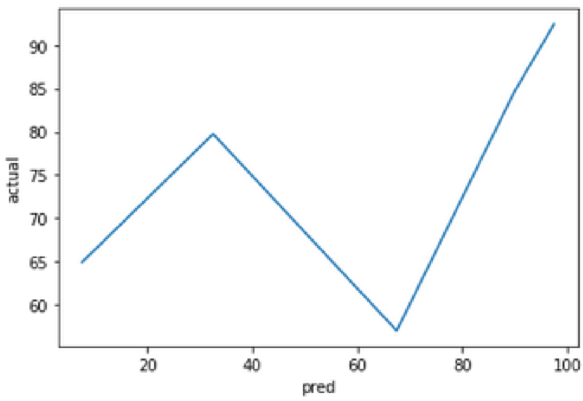


Figure 2.38: Actual probability vs the predicted probability of regression network (predicted character)

As you see, not only has this model increased the amount of data under 55% threshold, but it has also destroyed the credibility of other intervals and the curve is not ascendant anymore.

2.4.3 Kullback–Leibler divergence

The layers are constructed as before. Figure 2.39 shows the loss curve on the train and validation set. The loss equals 0.0919 on the train set and 0.0870 on the validation set. Accordingly, the optimum number of epochs is 30.

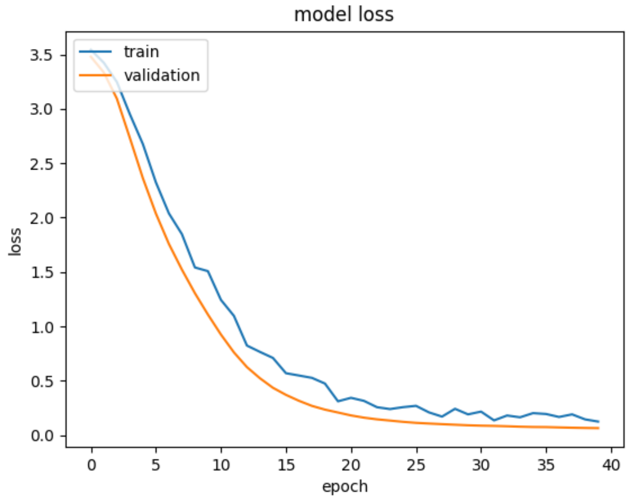
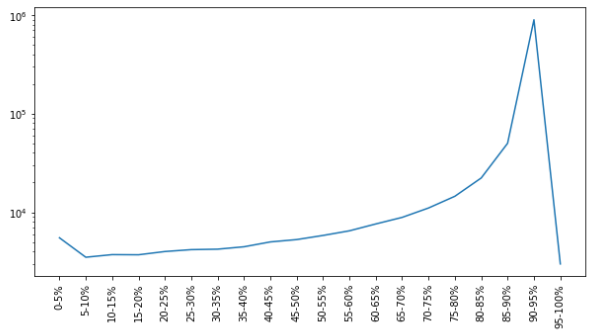
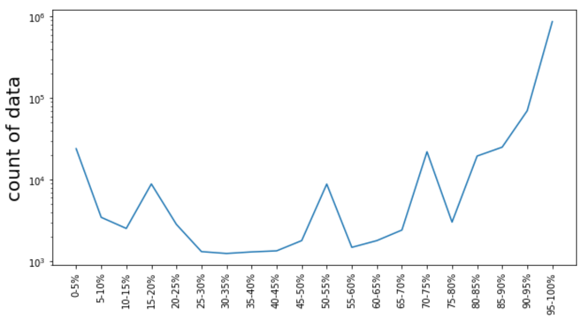


Figure 2.39: Loss of the model with KLD loss function

The loss on the test set is equal to 0.0931. The distribution of first best element in predicted vectors is shown below (2.40). As evident from figure 2.40, 5.37% of data exist under 55% threshold.



(a) Distribution before training



(b) Distribution after training

Figure 2.40: Comparison between the distribution of predicted character in the model with KLD loss function

This model has increased the amount of data under 55% threshold.

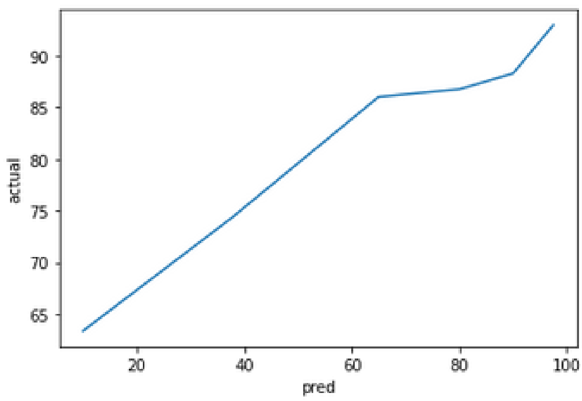
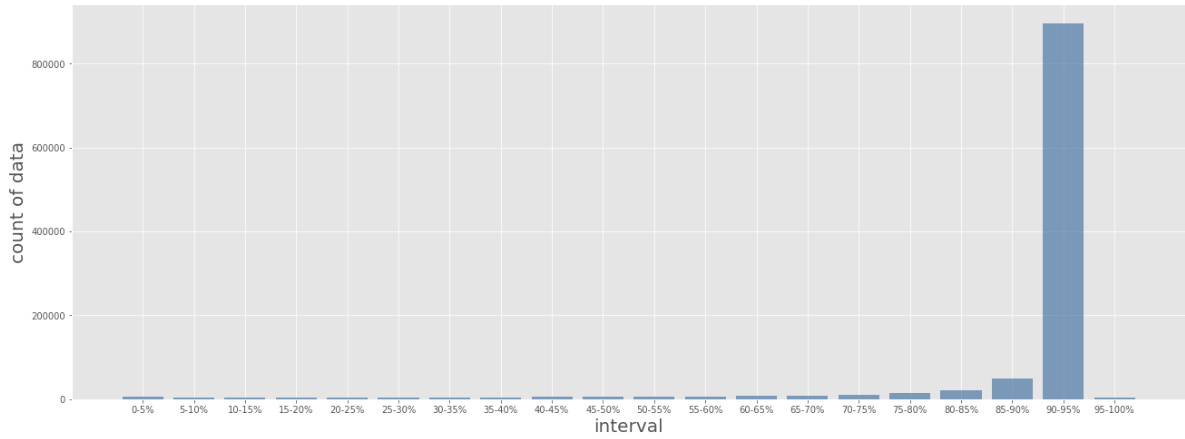


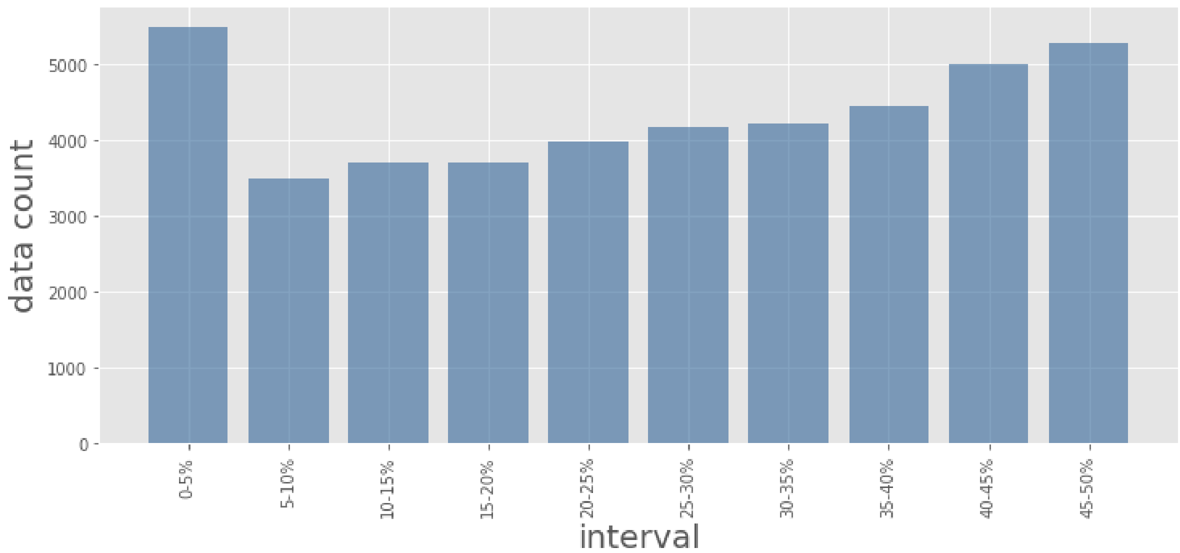
Figure 2.41: Actual probability vs the predicted probability of KLD network (predicted character)

2.5 Dataset Construction (Third approach)

Since the most of the problem stem from the first half of the data set meaning under 50%, we split the whole data set with the threshold of 50% and trained the network on first half.



(a) Distribution before reconstruction



(b) Distribution after reconstruction

Figure 2.42: Distribution comparison before and after reconstruction

2.5.1 Cross-entropy loss

Figure 2.43 shows the loss curve on the train and validation set. The loss equals 0.3538 on the train set and 0.3530 on the validation set. Accordingly, the optimum number of epochs is 70

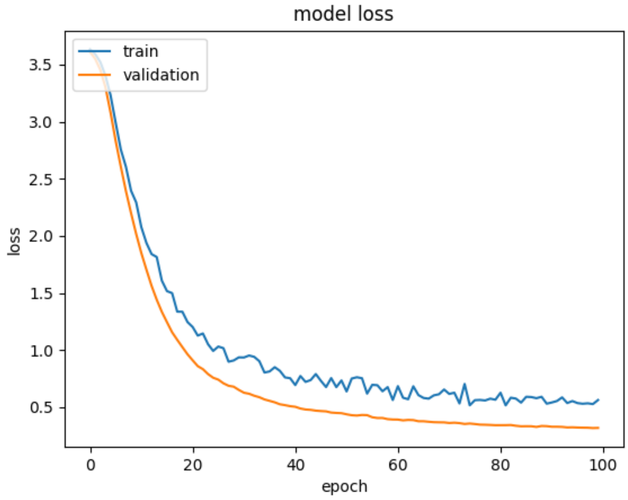
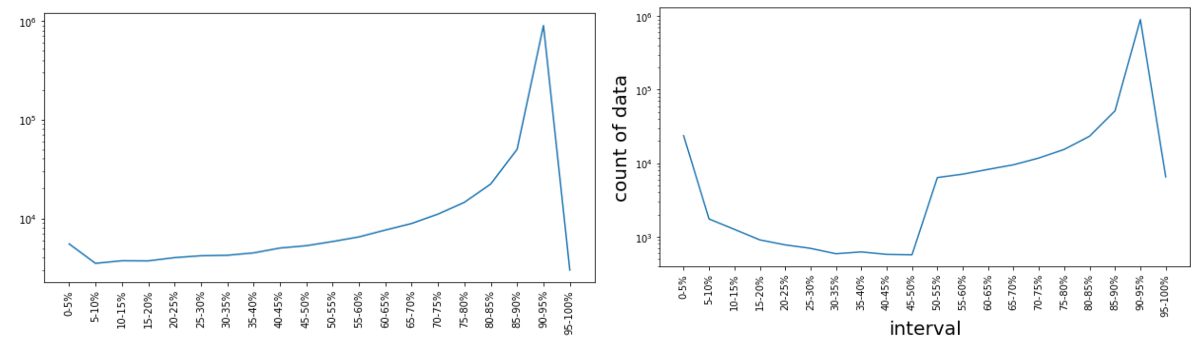


Figure 2.43: Loss of the model with Cross-entropy loss function

The loss on the test set is equal to 0.3609. The distribution of first best element in predicted vectors is shown below (2.44). As evident from figure 2.44, 3.53% of data exist under 55% threshold.



(a) Distribution before training

(b) Distribution after training

Figure 2.44: Comparison between the distribution of predicted character in the model with Cross-entropy loss function

Figure 2.45 shows the predicted versus actual probability of this network.

Not only has it decreased the data before 55% interval but it has also kept the ascending shape of the plot. We can see the behavior of this model on the other 36 options of each index in figure 2.46.

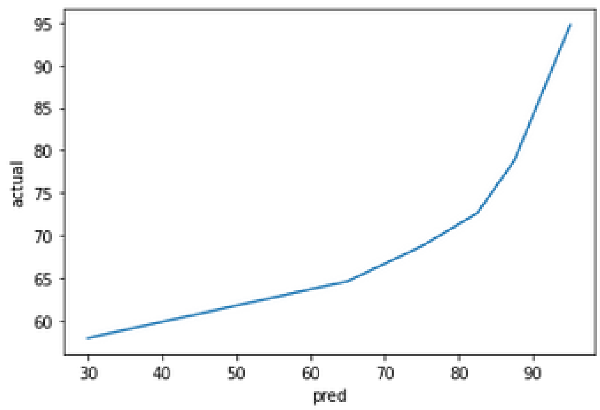
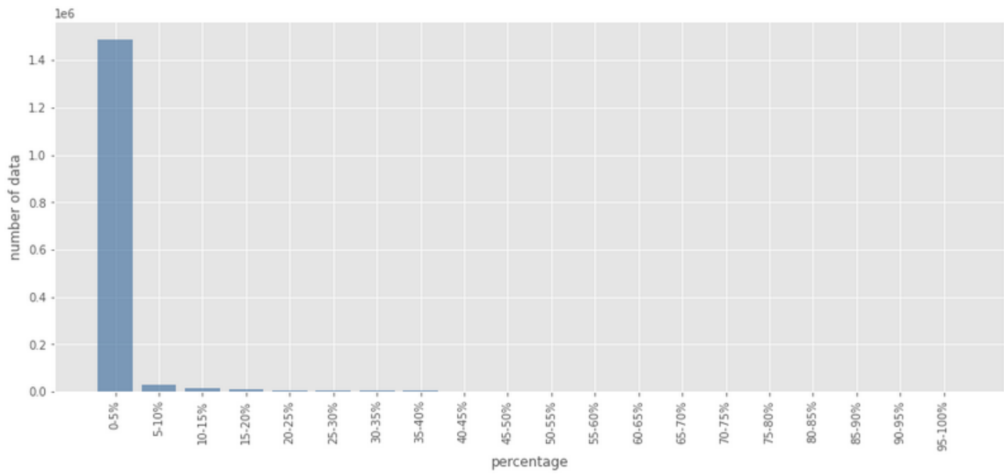
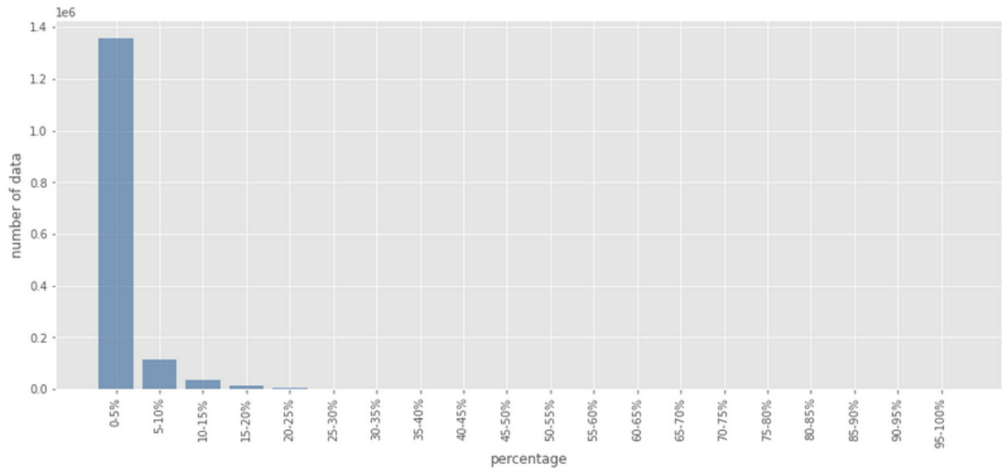


Figure 2.45: Actual probability vs the predicted probability of Cross-entropy network (predicted character)



(a) Distribution before training



(b) Distribution after training

Figure 2.46: Distribution comparison in the model with Cross-entropy loss function

Using table 2.2, we understand that all of the probabilities corresponding to other 36 character options should reside between 0 to 5% interval. Before training any network, we had 94.99% of data in this interval but as you can see in figure 2.46, after training some data has been transferred to other intervals and we have only 89.01% in this interval so the results have gotten worse.

2.5.2 Mean Square Error loss

Figure 2.47 shows the loss curve on the train and validation set. The loss equals 0.0030 and MAE equals 0.0075 on the train set and we have the loss equal to 0.0031 and MAE equal to 0.0078 on the validation set. Accordingly, the optimum number of epochs is 150.

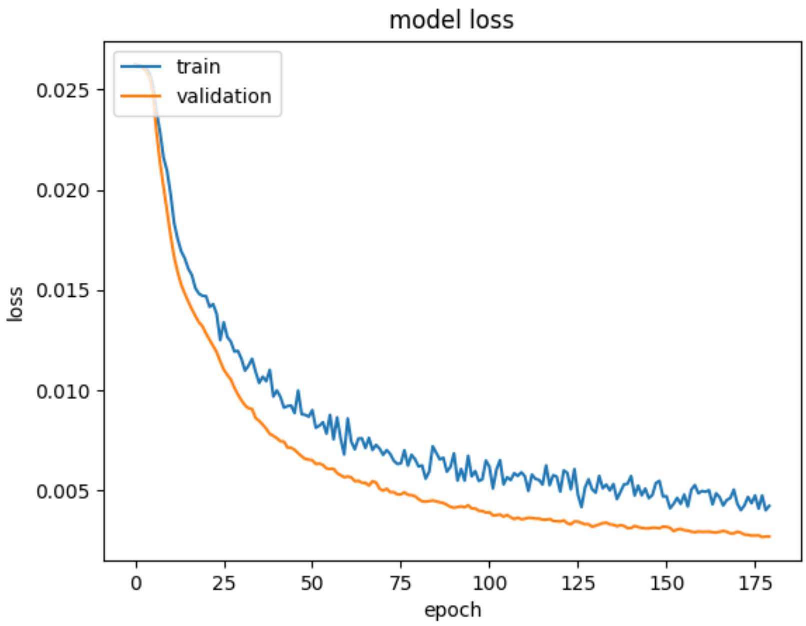
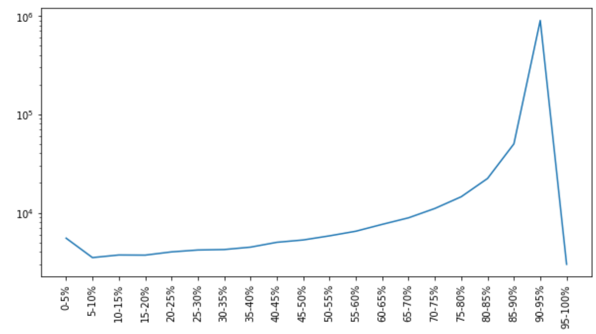


Figure 2.47: Loss of the model with MSE loss function

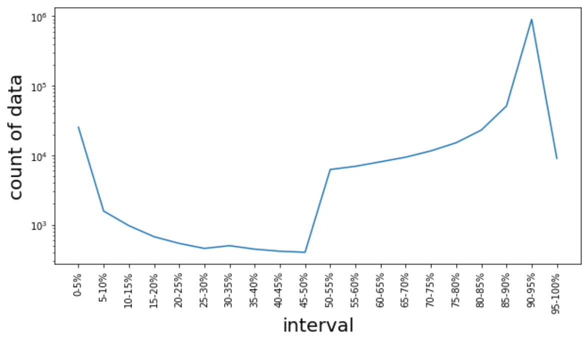
The loss is 0.0032 and the MAE is 0.0080 on the test set. The distribution of first best element in predicted vectors is shown below (2.48). As evident from figure 2.48, 3.50% of data exist under 55% threshold.

Figure 2.49 shows the predicted versus actual probability of this network.

Not only has it decreased the data before 55% interval but it has also kept the ascending shape of the plot. We can see the behavior of this model on the other 36 options of each index in figure 2.50.



(a) Distribution before training



(b) Distribution after training

Figure 2.48: Comparison between the distribution of predicted character in the model with MSE loss function

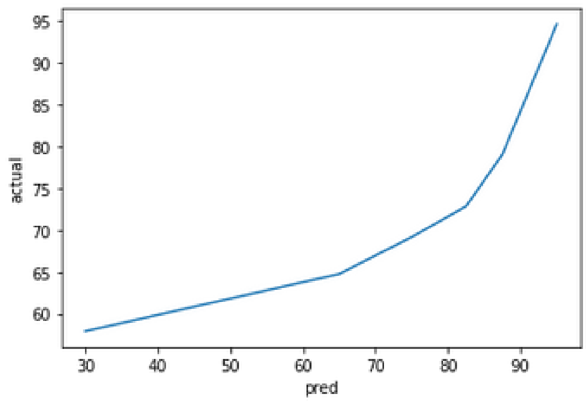
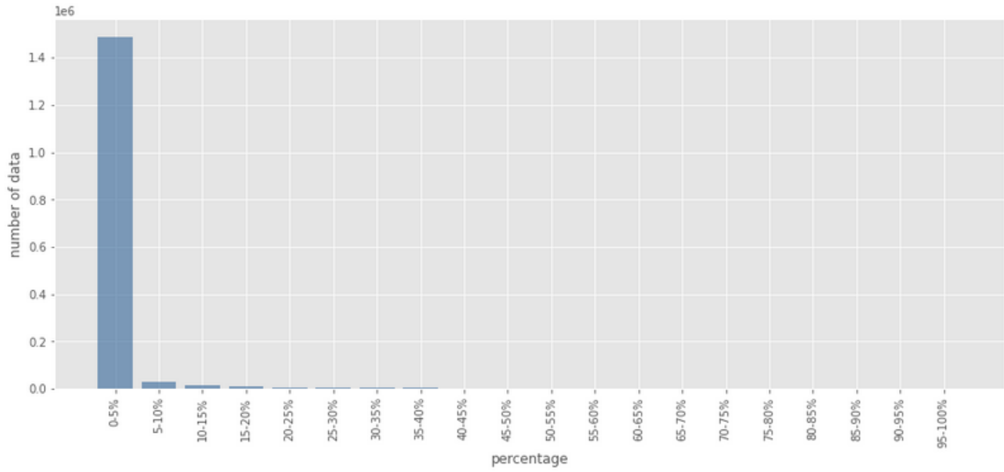
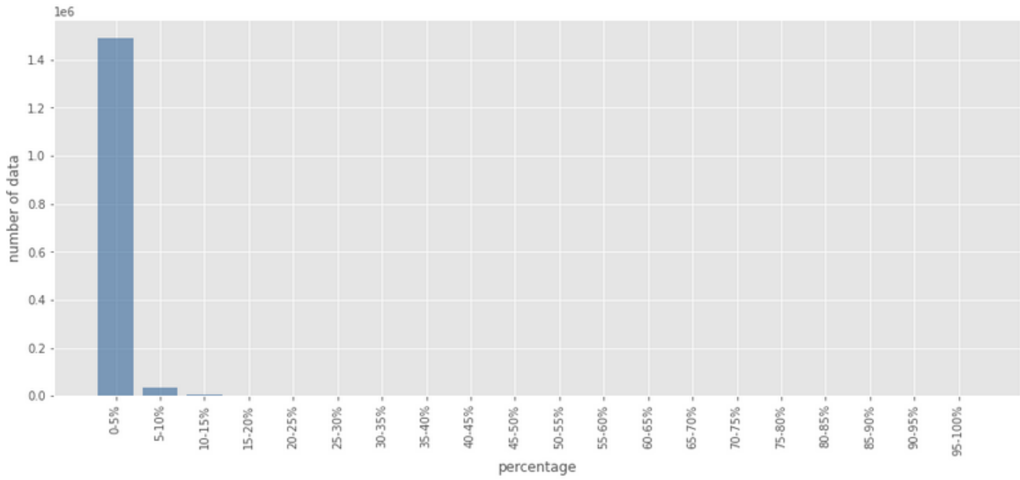


Figure 2.49: Actual probability vs the predicted probability of regression network (predicted character)

After training using MSE loss function, we have 96.98% of data in 0 to 5% interval which is an improvement.



(a) Distribution before training



(b) Distribution after training

Figure 2.50: Distribution comparison in the model with MSE loss function

2.5.3 Kullback–Leibler divergence

The models are constructed like before. Figure 2.51 shows the loss curve on the train and validation set. The loss equals 0.3309 on the train set and 0.3394 on the validation set. Accordingly, the optimum number of epochs is 50.

The loss on the test set is equal to 0.3394. The distribution of first best element in predicted vectors is shown below (2.52). As evident from figure 2.52, 3.63% of data exist under 55% threshold.

Figure 2.53 shows the predicted versus actual probability of this network.

Not only has it decreased the data before 55% interval but it has also kept the

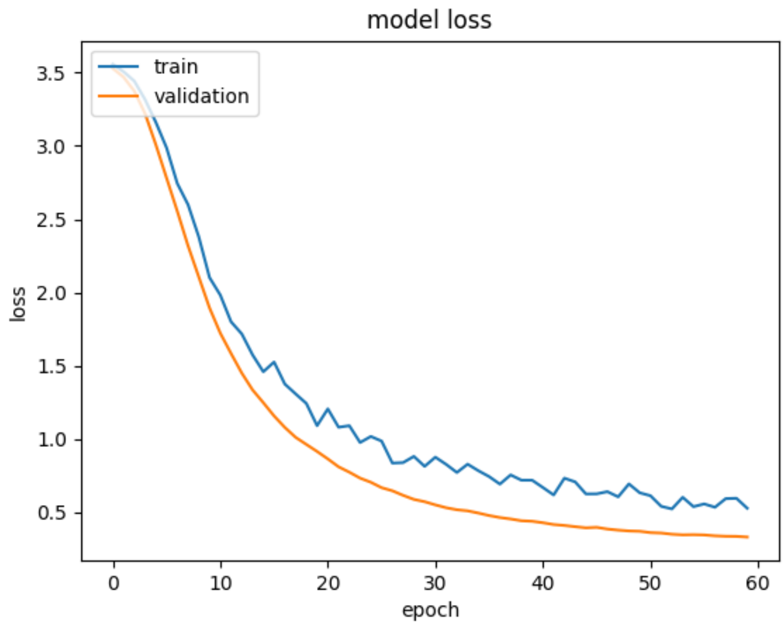
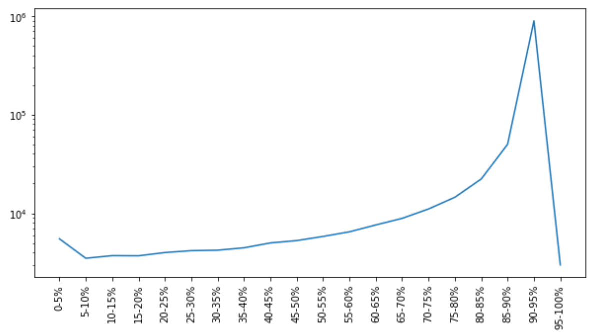
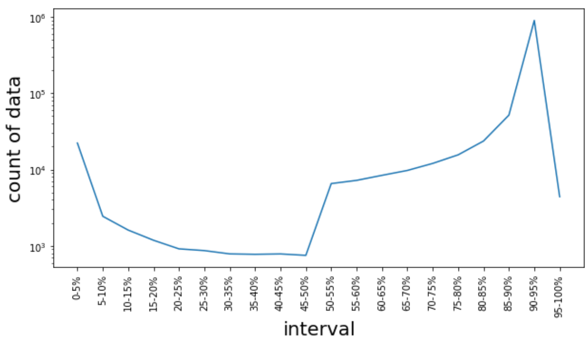


Figure 2.51: Loss of the model with KLD loss function



(a) Distribution before training



(b) Distribution after training

Figure 2.52: Comparison between the distribution of predicted character in the model with KLD loss function (Data Construction 3 section)

ascending shape of the plot. We can see the behavior of this model on the other 36 options of each index in figure 2.54.

After training using KLD loss function, we have 97.01% of data in 0 to 5% interval which is an improvement.

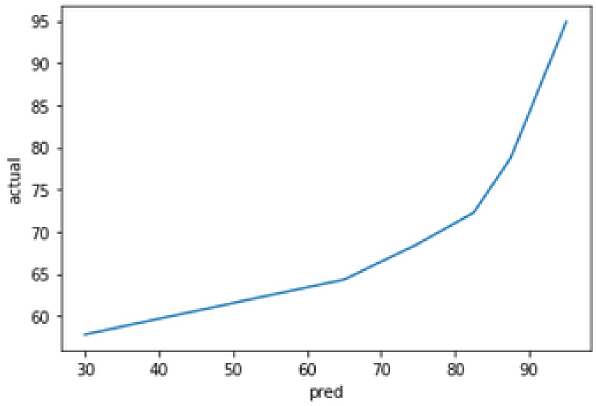
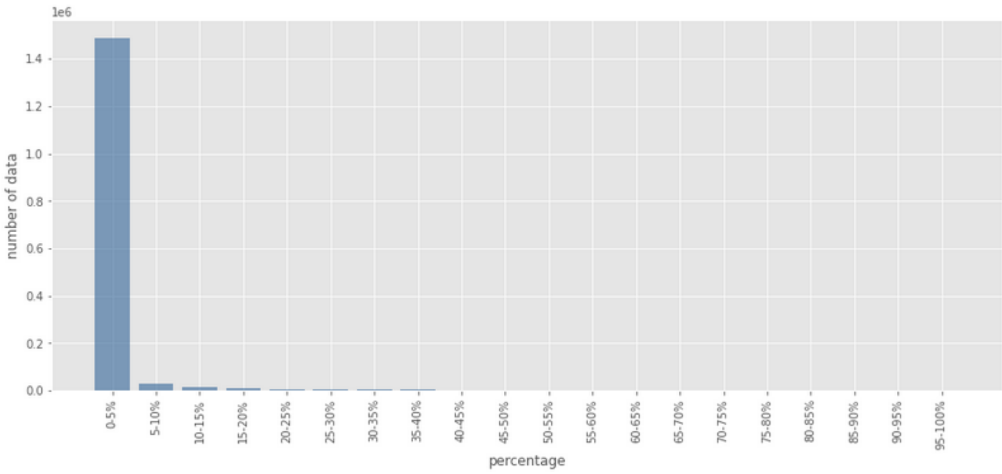
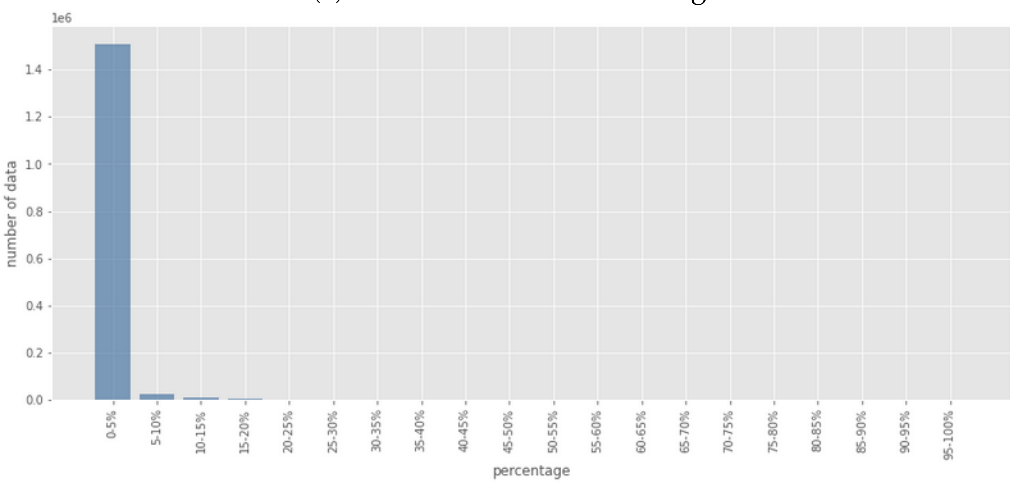


Figure 2.53: Actual probability vs the predicted probability of KLD network (preicted character)



(a) Distribution before training



(b) Distribution after training

Figure 2.54: Distribution comparison in the model with KLD loss function

3 Discussions and Conclusions

In conclusion, the best result we got was after training an MLP with KLD loss function on first half of the data. Its results can be found in section 2.5.3.

Bibliography

[1] The Spoken Dutch Corpus documentation.

[2] sclite - score speech recognition system output

A1 Appendix 1 - Intervals

In this python script, you can find the intervals which has equal amount of data. The dictionary's keys correspond to reversed rank of each probability in the vector (e.g. 0 corresponds to the 37th best probability).

```
intervals ={
0 :
[[8e-05, 0.00075], [0.00075, 0.00089], [0.00089, 0.00099], [0.00099,
    0.00107], [0.00107, 0.00114], [0.00114, 0.0012], [0.0012, 0.00126],
    [0.00126, 0.00131], [0.00131, 0.00136], [0.00136, 0.0014], [0.0014,
    0.00144], [0.00144, 0.00148], [0.00148, 0.00152], [0.00152,
    0.00155], [0.00155, 0.00159], [0.00159, 0.00162], [0.00162,
    0.00165], [0.00165, 0.00168], [0.00168, 0.00171], [0.00171,
    0.00174], [0.00174, 0.00177], [0.00177, 0.0018], [0.0018, 0.00183],
    [0.00183, 0.00185], [0.00185, 0.00188], [0.00188, 0.00191],
    [0.00191, 0.00195], [0.00195, 0.00199], [0.00199, 0.00204],
    [0.00204, 0.00504]] ,
1 :
[[9e-05, 0.00093], [0.00093, 0.00108], [0.00108, 0.00118], [0.00118,
    0.00125], [0.00125, 0.00132], [0.00132, 0.00138], [0.00138,
    0.00143], [0.00143, 0.00148], [0.00148, 0.00152], [0.00152,
    0.00156], [0.00156, 0.0016], [0.0016, 0.00163], [0.00163, 0.00167],
    [0.00167, 0.0017], [0.0017, 0.00173], [0.00173, 0.00176], [0.00176,
    0.00179], [0.00179, 0.00181], [0.00181, 0.00184], [0.00184,
    0.00186], [0.00186, 0.00189], [0.00189, 0.00191], [0.00191,
    0.00193], [0.00193, 0.00196], [0.00196, 0.00198], [0.00198,
```

```
    0.00201], [0.00201, 0.00203], [0.00203, 0.00207], [0.00207,
    0.00212], [0.00212, 0.00547]] ,
2 :
[[0.00016, 0.00103], [0.00103, 0.00118], [0.00118, 0.00128], [0.00128,
    0.00136], [0.00136, 0.00142], [0.00142, 0.00148], [0.00148,
    0.00153], [0.00153, 0.00157], [0.00157, 0.00161], [0.00161,
    0.00165], [0.00165, 0.00169], [0.00169, 0.00172], [0.00172,
    0.00175], [0.00175, 0.00178], [0.00178, 0.00181], [0.00181,
    0.00184], [0.00184, 0.00186], [0.00186, 0.00189], [0.00189,
    0.00191], [0.00191, 0.00193], [0.00193, 0.00195], [0.00195,
    0.00197], [0.00197, 0.00199], [0.00199, 0.00201], [0.00201,
    0.00203], [0.00203, 0.00205], [0.00205, 0.00208], [0.00208,
    0.00211], [0.00211, 0.00217], [0.00217, 0.00597]] ,
3 :
[[0.00016, 0.00111], [0.00111, 0.00126], [0.00126, 0.00135], [0.00135,
    0.00143], [0.00143, 0.00149], [0.00149, 0.00155], [0.00155,
    0.00159], [0.00159, 0.00164], [0.00164, 0.00168], [0.00168,
    0.00172], [0.00172, 0.00175], [0.00175, 0.00178], [0.00178,
    0.00181], [0.00181, 0.00184], [0.00184, 0.00187], [0.00187,
    0.00189], [0.00189, 0.00191], [0.00191, 0.00194], [0.00194,
    0.00196], [0.00196, 0.00198], [0.00198, 0.00199], [0.00199,
    0.00201], [0.00201, 0.00203], [0.00203, 0.00205], [0.00205,
    0.00207], [0.00207, 0.00209], [0.00209, 0.00211], [0.00211,
    0.00214], [0.00214, 0.00224], [0.00224, 0.00662]] ,
4 :
[[0.00017, 0.00117], [0.00117, 0.00132], [0.00132, 0.00141], [0.00141,
    0.00149], [0.00149, 0.00155], [0.00155, 0.0016], [0.0016, 0.00165],
    [0.00165, 0.00169], [0.00169, 0.00173], [0.00173, 0.00177],
    [0.00177, 0.0018], [0.0018, 0.00183], [0.00183, 0.00186], [0.00186,
    0.00188], [0.00188, 0.00191], [0.00191, 0.00193], [0.00193,
    0.00195], [0.00195, 0.00197], [0.00197, 0.00199], [0.00199,
    0.00201], [0.00201, 0.00203], [0.00203, 0.00204], [0.00204,
    0.00206], [0.00206, 0.00208], [0.00208, 0.0021], [0.0021, 0.00212],
    [0.00212, 0.00214], [0.00214, 0.00217], [0.00217, 0.00232],
    [0.00232, 0.00697]] ,
```

```
5 :
[[0.0002, 0.00122], [0.00122, 0.00137], [0.00137, 0.00146], [0.00146,
    0.00154], [0.00154, 0.0016], [0.0016, 0.00165], [0.00165, 0.0017],
    [0.0017, 0.00174], [0.00174, 0.00177], [0.00177, 0.00181], [0.00181,
     0.00184], [0.00184, 0.00187], [0.00187, 0.00189], [0.00189,
    0.00192], [0.00192, 0.00194], [0.00194, 0.00196], [0.00196,
    0.00198], [0.00198, 0.002], [0.002, 0.00202], [0.00202, 0.00204],
    [0.00204, 0.00205], [0.00205, 0.00207], [0.00207, 0.00209],
    [0.00209, 0.0021], [0.0021, 0.00212], [0.00212, 0.00214], [0.00214,
    0.00216], [0.00216, 0.00221], [0.00221, 0.0024], [0.0024, 0.0073]] ,
6 :
[[0.00022, 0.00127], [0.00127, 0.00141], [0.00141, 0.00151], [0.00151,
    0.00158], [0.00158, 0.00164], [0.00164, 0.00169], [0.00169,
    0.00173], [0.00173, 0.00177], [0.00177, 0.00181], [0.00181,
    0.00184], [0.00184, 0.00187], [0.00187, 0.0019], [0.0019, 0.00193],
    [0.00193, 0.00195], [0.00195, 0.00197], [0.00197, 0.00199],
    [0.00199, 0.00201], [0.00201, 0.00203], [0.00203, 0.00205],
    [0.00205, 0.00206], [0.00206, 0.00208], [0.00208, 0.00209],
    [0.00209, 0.00211], [0.00211, 0.00212], [0.00212, 0.00214],
    [0.00214, 0.00216], [0.00216, 0.00218], [0.00218, 0.00224],
    [0.00224, 0.00248], [0.00248, 0.00787]] ,
7 :
[[0.00023, 0.00131], [0.00131, 0.00145], [0.00145, 0.00155], [0.00155,
    0.00162], [0.00162, 0.00168], [0.00168, 0.00173], [0.00173,
    0.00177], [0.00177, 0.00181], [0.00181, 0.00184], [0.00184,
    0.00187], [0.00187, 0.0019], [0.0019, 0.00193], [0.00193, 0.00195],
    [0.00195, 0.00198], [0.00198, 0.002], [0.002, 0.00202], [0.00202,
    0.00203], [0.00203, 0.00205], [0.00205, 0.00207], [0.00207,
    0.00208], [0.00208, 0.0021], [0.0021, 0.00211], [0.00211, 0.00213],
    [0.00213, 0.00214], [0.00214, 0.00216], [0.00216, 0.00218],
    [0.00218, 0.00221], [0.00221, 0.00228], [0.00228, 0.00255],
    [0.00255, 0.00814]] ,
8 :
[[0.00027, 0.00135], [0.00135, 0.00149], [0.00149, 0.00158], [0.00158,
    0.00166], [0.00166, 0.00171], [0.00171, 0.00176], [0.00176, 0.0018],
```

```
 [0.0018, 0.00184], [0.00184, 0.00187], [0.00187, 0.0019], [0.0019,
 0.00193], [0.00193, 0.00196], [0.00196, 0.00198], [0.00198, 0.002],
 [0.002, 0.00202], [0.00202, 0.00204], [0.00204, 0.00205], [0.00205,
 0.00207], [0.00207, 0.00209], [0.00209, 0.0021], [0.0021, 0.00211],
 [0.00211, 0.00213], [0.00213, 0.00214], [0.00214, 0.00216],
 [0.00216, 0.00217], [0.00217, 0.00219], [0.00219, 0.00223],
 [0.00223, 0.00232], [0.00232, 0.00261], [0.00261, 0.00879]] ,
9 :
[[0.00027, 0.00138], [0.00138, 0.00153], [0.00153, 0.00162], [0.00162,
 0.00169], [0.00169, 0.00174], [0.00174, 0.00179], [0.00179,
 0.00183], [0.00183, 0.00187], [0.00187, 0.0019], [0.0019, 0.00193],
 [0.00193, 0.00196], [0.00196, 0.00198], [0.00198, 0.002], [0.002,
 0.00202], [0.00202, 0.00204], [0.00204, 0.00206], [0.00206,
 0.00207], [0.00207, 0.00209], [0.00209, 0.0021], [0.0021, 0.00212],
 [0.00212, 0.00213], [0.00213, 0.00214], [0.00214, 0.00216],
 [0.00216, 0.00217], [0.00217, 0.00219], [0.00219, 0.00221],
 [0.00221, 0.00225], [0.00225, 0.00235], [0.00235, 0.00268],
 [0.00268, 0.00928]] ,
10 :
[[0.0003, 0.00142], [0.00142, 0.00156], [0.00156, 0.00165], [0.00165,
 0.00172], [0.00172, 0.00177], [0.00177, 0.00182], [0.00182,
 0.00186], [0.00186, 0.00189], [0.00189, 0.00192], [0.00192,
 0.00195], [0.00195, 0.00198], [0.00198, 0.002], [0.002, 0.00202],
 [0.00202, 0.00204], [0.00204, 0.00206], [0.00206, 0.00207],
 [0.00207, 0.00209], [0.00209, 0.0021], [0.0021, 0.00212], [0.00212,
 0.00213], [0.00213, 0.00214], [0.00214, 0.00216], [0.00216,
 0.00217], [0.00217, 0.00218], [0.00218, 0.0022], [0.0022, 0.00223],
 [0.00223, 0.00228], [0.00228, 0.00239], [0.00239, 0.00274],
 [0.00274, 0.01013]] ,
11 :
[[0.00034, 0.00145], [0.00145, 0.00159], [0.00159, 0.00168], [0.00168,
 0.00175], [0.00175, 0.0018], [0.0018, 0.00184], [0.00184, 0.00188],
 [0.00188, 0.00192], [0.00192, 0.00195], [0.00195, 0.00197],
 [0.00197, 0.002], [0.002, 0.00202], [0.00202, 0.00204], [0.00204,
 0.00206], [0.00206, 0.00207], [0.00207, 0.00209], [0.00209, 0.0021],
```

```
 [0.0021, 0.00212], [0.00212, 0.00213], [0.00213, 0.00214],
[0.00214, 0.00216], [0.00216, 0.00217], [0.00217, 0.00218],
[0.00218, 0.0022], [0.0022, 0.00222], [0.00222, 0.00225], [0.00225,
0.0023], [0.0023, 0.00243], [0.00243, 0.0028], [0.0028, 0.01067]] ,
12 :
[[0.00034, 0.00148], [0.00148, 0.00162], [0.00162, 0.00171], [0.00171,
0.00177], [0.00177, 0.00182], [0.00182, 0.00187], [0.00187,
0.00191], [0.00191, 0.00194], [0.00194, 0.00197], [0.00197,
0.00199], [0.00199, 0.00202], [0.00202, 0.00204], [0.00204,
0.00206], [0.00206, 0.00207], [0.00207, 0.00209], [0.00209, 0.0021],
 [0.0021, 0.00212], [0.00212, 0.00213], [0.00213, 0.00214],
[0.00214, 0.00215], [0.00215, 0.00217], [0.00217, 0.00218],
[0.00218, 0.00219], [0.00219, 0.00221], [0.00221, 0.00223],
[0.00223, 0.00227], [0.00227, 0.00233], [0.00233, 0.00247],
[0.00247, 0.00286], [0.00286, 0.01104]] ,
13 :
[[0.00036, 0.00151], [0.00151, 0.00165], [0.00165, 0.00173], [0.00173,
0.0018], [0.0018, 0.00185], [0.00185, 0.00189], [0.00189, 0.00193],
[0.00193, 0.00196], [0.00196, 0.00199], [0.00199, 0.00201],
[0.00201, 0.00203], [0.00203, 0.00205], [0.00205, 0.00207],
[0.00207, 0.00209], [0.00209, 0.0021], [0.0021, 0.00212], [0.00212,
0.00213], [0.00213, 0.00214], [0.00214, 0.00215], [0.00215,
0.00217], [0.00217, 0.00218], [0.00218, 0.00219], [0.00219,
0.00221], [0.00221, 0.00222], [0.00222, 0.00225], [0.00225,
0.00229], [0.00229, 0.00235], [0.00235, 0.00251], [0.00251,
0.00292], [0.00292, 0.01195]] ,
14 :
[[0.00037, 0.00153], [0.00153, 0.00167], [0.00167, 0.00176], [0.00176,
0.00182], [0.00182, 0.00187], [0.00187, 0.00191], [0.00191,
0.00195], [0.00195, 0.00198], [0.00198, 0.002], [0.002, 0.00203],
[0.00203, 0.00205], [0.00205, 0.00207], [0.00207, 0.00209],
[0.00209, 0.0021], [0.0021, 0.00212], [0.00212, 0.00213], [0.00213,
0.00214], [0.00214, 0.00215], [0.00215, 0.00217], [0.00217,
0.00218], [0.00218, 0.00219], [0.00219, 0.0022], [0.0022, 0.00222],
[0.00222, 0.00224], [0.00224, 0.00226], [0.00226, 0.00231],
```

```
    [0.00231, 0.00238], [0.00238, 0.00255], [0.00255, 0.00298],
    [0.00298, 0.01253]] ,
15 :
[[0.0004, 0.00156], [0.00156, 0.00169], [0.00169, 0.00178], [0.00178,
    0.00184], [0.00184, 0.00189], [0.00189, 0.00193], [0.00193,
    0.00196], [0.00196, 0.00199], [0.00199, 0.00202], [0.00202,
    0.00204], [0.00204, 0.00206], [0.00206, 0.00208], [0.00208, 0.0021],
     [0.0021, 0.00211], [0.00211, 0.00213], [0.00213, 0.00214],
    [0.00214, 0.00215], [0.00215, 0.00216], [0.00216, 0.00218],
    [0.00218, 0.00219], [0.00219, 0.0022], [0.0022, 0.00221], [0.00221,
    0.00223], [0.00223, 0.00225], [0.00225, 0.00228], [0.00228,
    0.00232], [0.00232, 0.0024], [0.0024, 0.00259], [0.00259, 0.00304],
    [0.00304, 0.01359]] ,
16 :
[[0.00041, 0.00158], [0.00158, 0.00172], [0.00172, 0.0018], [0.0018,
    0.00186], [0.00186, 0.00191], [0.00191, 0.00195], [0.00195,
    0.00198], [0.00198, 0.00201], [0.00201, 0.00204], [0.00204,
    0.00206], [0.00206, 0.00208], [0.00208, 0.00209], [0.00209,
    0.00211], [0.00211, 0.00212], [0.00212, 0.00214], [0.00214,
    0.00215], [0.00215, 0.00216], [0.00216, 0.00217], [0.00217,
    0.00219], [0.00219, 0.0022], [0.0022, 0.00221], [0.00221, 0.00222],
    [0.00222, 0.00224], [0.00224, 0.00226], [0.00226, 0.0023], [0.0023,
    0.00234], [0.00234, 0.00243], [0.00243, 0.00263], [0.00263, 0.0031],
     [0.0031, 0.01488]] ,
17 :
[[0.00042, 0.0016], [0.0016, 0.00174], [0.00174, 0.00182], [0.00182,
    0.00188], [0.00188, 0.00193], [0.00193, 0.00196], [0.00196, 0.002],
    [0.002, 0.00202], [0.00202, 0.00205], [0.00205, 0.00207], [0.00207,
    0.00209], [0.00209, 0.00211], [0.00211, 0.00212], [0.00212,
    0.00213], [0.00213, 0.00215], [0.00215, 0.00216], [0.00216,
    0.00217], [0.00217, 0.00218], [0.00218, 0.00219], [0.00219,
    0.00221], [0.00221, 0.00222], [0.00222, 0.00223], [0.00223,
    0.00225], [0.00225, 0.00228], [0.00228, 0.00231], [0.00231,
    0.00236], [0.00236, 0.00245], [0.00245, 0.00267], [0.00267,
    0.00317], [0.00317, 0.01549]] ,
```

```
18 :
[[0.00042, 0.00162], [0.00162, 0.00176], [0.00176, 0.00184], [0.00184,
    0.0019], [0.0019, 0.00194], [0.00194, 0.00198], [0.00198, 0.00201],
    [0.00201, 0.00204], [0.00204, 0.00206], [0.00206, 0.00208],
    [0.00208, 0.0021], [0.0021, 0.00212], [0.00212, 0.00213], [0.00213,
    0.00214], [0.00214, 0.00216], [0.00216, 0.00217], [0.00217,
    0.00218], [0.00218, 0.00219], [0.00219, 0.0022], [0.0022, 0.00222],
    [0.00222, 0.00223], [0.00223, 0.00224], [0.00224, 0.00226],
    [0.00226, 0.00229], [0.00229, 0.00233], [0.00233, 0.00238],
    [0.00238, 0.00248], [0.00248, 0.00271], [0.00271, 0.00323],
    [0.00323, 0.01675]] ,
19 :
[[0.00043, 0.00164], [0.00164, 0.00177], [0.00177, 0.00185], [0.00185,
    0.00191], [0.00191, 0.00196], [0.00196, 0.00199], [0.00199,
    0.00202], [0.00202, 0.00205], [0.00205, 0.00207], [0.00207,
    0.00209], [0.00209, 0.00211], [0.00211, 0.00213], [0.00213,
    0.00214], [0.00214, 0.00215], [0.00215, 0.00217], [0.00217,
    0.00218], [0.00218, 0.00219], [0.00219, 0.0022], [0.0022, 0.00221],
    [0.00221, 0.00222], [0.00222, 0.00224], [0.00224, 0.00225],
    [0.00225, 0.00228], [0.00228, 0.0023], [0.0023, 0.00234], [0.00234,
    0.0024], [0.0024, 0.00251], [0.00251, 0.00275], [0.00275, 0.0033],
    [0.0033, 0.0188]] ,
20 :
[[0.00043, 0.00166], [0.00166, 0.00179], [0.00179, 0.00187], [0.00187,
    0.00193], [0.00193, 0.00197], [0.00197, 0.00201], [0.00201,
    0.00204], [0.00204, 0.00206], [0.00206, 0.00208], [0.00208, 0.0021],
     [0.0021, 0.00212], [0.00212, 0.00214], [0.00214, 0.00215],
    [0.00215, 0.00216], [0.00216, 0.00217], [0.00217, 0.00219],
    [0.00219, 0.0022], [0.0022, 0.00221], [0.00221, 0.00222], [0.00222,
    0.00223], [0.00223, 0.00225], [0.00225, 0.00227], [0.00227,
    0.00229], [0.00229, 0.00232], [0.00232, 0.00236], [0.00236,
    0.00242], [0.00242, 0.00253], [0.00253, 0.00279], [0.00279,
    0.00338], [0.00338, 0.02044]] ,
21 :
```

```
[[0.00043, 0.00167], [0.00167, 0.00181], [0.00181, 0.00189], [0.00189,
    0.00194], [0.00194, 0.00198], [0.00198, 0.00202], [0.00202,
    0.00205], [0.00205, 0.00207], [0.00207, 0.00209], [0.00209,
    0.00211], [0.00211, 0.00213], [0.00213, 0.00214], [0.00214,
    0.00216], [0.00216, 0.00217], [0.00217, 0.00218], [0.00218,
    0.00219], [0.00219, 0.0022], [0.0022, 0.00222], [0.00222, 0.00223],
    [0.00223, 0.00224], [0.00224, 0.00226], [0.00226, 0.00228],
    [0.00228, 0.0023], [0.0023, 0.00233], [0.00233, 0.00237], [0.00237,
    0.00244], [0.00244, 0.00256], [0.00256, 0.00284], [0.00284,
    0.00346], [0.00346, 0.02293]] ,
22 :
[[0.00043, 0.00169], [0.00169, 0.00182], [0.00182, 0.0019], [0.0019,
    0.00196], [0.00196, 0.002], [0.002, 0.00203], [0.00203, 0.00206],
    [0.00206, 0.00208], [0.00208, 0.0021], [0.0021, 0.00212], [0.00212,
    0.00214], [0.00214, 0.00215], [0.00215, 0.00217], [0.00217,
    0.00218], [0.00218, 0.00219], [0.00219, 0.0022], [0.0022, 0.00221],
    [0.00221, 0.00222], [0.00222, 0.00224], [0.00224, 0.00225],
    [0.00225, 0.00227], [0.00227, 0.00229], [0.00229, 0.00231],
    [0.00231, 0.00234], [0.00234, 0.00239], [0.00239, 0.00246],
    [0.00246, 0.00259], [0.00259, 0.00289], [0.00289, 0.00356],
    [0.00356, 0.02438]] ,
23 :
[[0.00045, 0.00171], [0.00171, 0.00184], [0.00184, 0.00192], [0.00192,
    0.00197], [0.00197, 0.00201], [0.00201, 0.00205], [0.00205,
    0.00207], [0.00207, 0.0021], [0.0021, 0.00212], [0.00212, 0.00213],
    [0.00213, 0.00215], [0.00215, 0.00216], [0.00216, 0.00217],
    [0.00217, 0.00219], [0.00219, 0.0022], [0.0022, 0.00221], [0.00221,
    0.00222], [0.00222, 0.00223], [0.00223, 0.00224], [0.00224,
    0.00226], [0.00226, 0.00228], [0.00228, 0.0023], [0.0023, 0.00232],
    [0.00232, 0.00236], [0.00236, 0.00241], [0.00241, 0.00248],
    [0.00248, 0.00262], [0.00262, 0.00295], [0.00295, 0.00368],
    [0.00368, 0.02722]] ,
24 :
[[0.00045, 0.00174], [0.00174, 0.00186], [0.00186, 0.00194], [0.00194,
    0.00199], [0.00199, 0.00203], [0.00203, 0.00206], [0.00206,
```

```
    0.00209], [0.00209, 0.00211], [0.00211, 0.00213], [0.00213,
    0.00214], [0.00214, 0.00216], [0.00216, 0.00217], [0.00217,
    0.00218], [0.00218, 0.00219], [0.00219, 0.00221], [0.00221,
    0.00222], [0.00222, 0.00223], [0.00223, 0.00224], [0.00224,
    0.00225], [0.00225, 0.00227], [0.00227, 0.00229], [0.00229,
    0.00231], [0.00231, 0.00234], [0.00234, 0.00238], [0.00238,
    0.00243], [0.00243, 0.00251], [0.00251, 0.00266], [0.00266,
    0.00303], [0.00303, 0.00383], [0.00383, 0.03078]] ,
25 :
[[0.00047, 0.00176], [0.00176, 0.00188], [0.00188, 0.00195], [0.00195,
    0.002], [0.002, 0.00204], [0.00204, 0.00207], [0.00207, 0.0021],
    [0.0021, 0.00212], [0.00212, 0.00214], [0.00214, 0.00215], [0.00215,
     0.00217], [0.00217, 0.00218], [0.00218, 0.00219], [0.00219,
    0.0022], [0.0022, 0.00222], [0.00222, 0.00223], [0.00223, 0.00224],
    [0.00224, 0.00225], [0.00225, 0.00226], [0.00226, 0.00228],
    [0.00228, 0.0023], [0.0023, 0.00232], [0.00232, 0.00235], [0.00235,
    0.00239], [0.00239, 0.00245], [0.00245, 0.00254], [0.00254,
    0.00271], [0.00271, 0.00312], [0.00312, 0.00402], [0.00402,
    0.03468]] ,
26 :
[[0.00049, 0.00179], [0.00179, 0.00191], [0.00191, 0.00198], [0.00198,
    0.00202], [0.00202, 0.00206], [0.00206, 0.00209], [0.00209,
    0.00211], [0.00211, 0.00213], [0.00213, 0.00215], [0.00215,
    0.00217], [0.00217, 0.00218], [0.00218, 0.00219], [0.00219, 0.0022],
     [0.0022, 0.00221], [0.00221, 0.00223], [0.00223, 0.00224],
    [0.00224, 0.00225], [0.00225, 0.00226], [0.00226, 0.00228],
    [0.00228, 0.00229], [0.00229, 0.00231], [0.00231, 0.00234],
    [0.00234, 0.00237], [0.00237, 0.00242], [0.00242, 0.00248],
    [0.00248, 0.00258], [0.00258, 0.00277], [0.00277, 0.00324],
    [0.00324, 0.00426], [0.00426, 0.03848]] ,
27 :
[[0.0005, 0.00182], [0.00182, 0.00193], [0.00193, 0.002], [0.002,
    0.00205], [0.00205, 0.00208], [0.00208, 0.00211], [0.00211,
    0.00213], [0.00213, 0.00215], [0.00215, 0.00216], [0.00216,
    0.00218], [0.00218, 0.00219], [0.00219, 0.0022], [0.0022, 0.00222],
```

```
[0.00222, 0.00223], [0.00223, 0.00224], [0.00224, 0.00225],
[0.00225, 0.00226], [0.00226, 0.00228], [0.00228, 0.00229],
[0.00229, 0.00231], [0.00231, 0.00233], [0.00233, 0.00236],
[0.00236, 0.0024], [0.0024, 0.00244], [0.00244, 0.00251], [0.00251,
0.00262], [0.00262, 0.00286], [0.00286, 0.00338], [0.00338, 0.0046],
[0.0046, 0.04364]] ,
28 :
[[0.00051, 0.00186], [0.00186, 0.00197], [0.00197, 0.00203], [0.00203,
0.00207], [0.00207, 0.0021], [0.0021, 0.00213], [0.00213, 0.00215],
[0.00215, 0.00217], [0.00217, 0.00218], [0.00218, 0.00219],
[0.00219, 0.00221], [0.00221, 0.00222], [0.00222, 0.00223],
[0.00223, 0.00224], [0.00224, 0.00225], [0.00225, 0.00226],
[0.00226, 0.00228], [0.00228, 0.00229], [0.00229, 0.00231],
[0.00231, 0.00233], [0.00233, 0.00236], [0.00236, 0.00239],
[0.00239, 0.00243], [0.00243, 0.00248], [0.00248, 0.00256],
[0.00256, 0.00269], [0.00269, 0.00298], [0.00298, 0.00358],
[0.00358, 0.00506], [0.00506, 0.05092]] ,
29 :
[[0.00052, 0.0019], [0.0019, 0.002], [0.002, 0.00206], [0.00206,
0.0021], [0.0021, 0.00213], [0.00213, 0.00215], [0.00215, 0.00217],
[0.00217, 0.00219], [0.00219, 0.0022], [0.0022, 0.00221], [0.00221,
0.00222], [0.00222, 0.00224], [0.00224, 0.00225], [0.00225,
0.00226], [0.00226, 0.00227], [0.00227, 0.00228], [0.00228, 0.0023],
[0.0023, 0.00231], [0.00231, 0.00233], [0.00233, 0.00236],
[0.00236, 0.00239], [0.00239, 0.00242], [0.00242, 0.00247],
[0.00247, 0.00253], [0.00253, 0.00262], [0.00262, 0.00278],
[0.00278, 0.00314], [0.00314, 0.00385], [0.00385, 0.00576],
[0.00576, 0.0608]] ,
30 :
[[0.00056, 0.00195], [0.00195, 0.00204], [0.00204, 0.0021], [0.0021,
0.00213], [0.00213, 0.00216], [0.00216, 0.00218], [0.00218,
0.00219], [0.00219, 0.00221], [0.00221, 0.00222], [0.00222,
0.00223], [0.00223, 0.00225], [0.00225, 0.00226], [0.00226,
0.00227], [0.00227, 0.00228], [0.00228, 0.00229], [0.00229,
0.00231], [0.00231, 0.00233], [0.00233, 0.00234], [0.00234,
```

```
    0.00237], [0.00237, 0.00239], [0.00239, 0.00243], [0.00243,
    0.00247], [0.00247, 0.00252], [0.00252, 0.0026], [0.0026, 0.00271],
    [0.00271, 0.00293], [0.00293, 0.00338], [0.00338, 0.00425],
    [0.00425, 0.00683], [0.00683, 0.07543]] ,
31 :
[[0.00063, 0.00201], [0.00201, 0.00209], [0.00209, 0.00214], [0.00214,
    0.00217], [0.00217, 0.00219], [0.00219, 0.00221], [0.00221,
    0.00222], [0.00222, 0.00224], [0.00224, 0.00225], [0.00225,
    0.00226], [0.00226, 0.00227], [0.00227, 0.00228], [0.00228, 0.0023],
     [0.0023, 0.00231], [0.00231, 0.00233], [0.00233, 0.00234],
    [0.00234, 0.00236], [0.00236, 0.00239], [0.00239, 0.00241],
    [0.00241, 0.00245], [0.00245, 0.00249], [0.00249, 0.00254],
    [0.00254, 0.00261], [0.00261, 0.0027], [0.0027, 0.00286], [0.00286,
    0.00317], [0.00317, 0.00376], [0.00376, 0.00491], [0.00491,
    0.00858], [0.00858, 0.09084]] ,
32 :
[[0.00067, 0.00207], [0.00207, 0.00215], [0.00215, 0.00218], [0.00218,
    0.00221], [0.00221, 0.00223], [0.00223, 0.00224], [0.00224,
    0.00226], [0.00226, 0.00227], [0.00227, 0.00228], [0.00228,
    0.00229], [0.00229, 0.00231], [0.00231, 0.00232], [0.00232,
    0.00234], [0.00234, 0.00236], [0.00236, 0.00238], [0.00238, 0.0024],
     [0.0024, 0.00242], [0.00242, 0.00245], [0.00245, 0.00249],
    [0.00249, 0.00253], [0.00253, 0.00259], [0.00259, 0.00266],
    [0.00266, 0.00275], [0.00275, 0.00289], [0.00289, 0.00314],
    [0.00314, 0.00359], [0.00359, 0.0044], [0.0044, 0.00615], [0.00615,
    0.01193], [0.01193, 0.12602]] ,
33 :
[[0.00074, 0.00215], [0.00215, 0.0022], [0.0022, 0.00223], [0.00223,
    0.00225], [0.00225, 0.00227], [0.00227, 0.00229], [0.00229, 0.0023],
     [0.0023, 0.00231], [0.00231, 0.00233], [0.00233, 0.00235],
    [0.00235, 0.00236], [0.00236, 0.00238], [0.00238, 0.00241],
    [0.00241, 0.00243], [0.00243, 0.00246], [0.00246, 0.00249],
    [0.00249, 0.00253], [0.00253, 0.00257], [0.00257, 0.00262],
    [0.00262, 0.00269], [0.00269, 0.00277], [0.00277, 0.00287],
    [0.00287, 0.00303], [0.00303, 0.00329], [0.00329, 0.00372],
```

```
    [0.00372, 0.00446], [0.00446, 0.00583], [0.00583, 0.00906],
    [0.00906, 0.01987], [0.01987, 0.17145]] ,
34 :
[[0.00099, 0.00223], [0.00223, 0.00227], [0.00227, 0.00229], [0.00229,
    0.00231], [0.00231, 0.00233], [0.00233, 0.00235], [0.00235,
    0.00237], [0.00237, 0.00239], [0.00239, 0.00242], [0.00242,
    0.00244], [0.00244, 0.00247], [0.00247, 0.0025], [0.0025, 0.00254],
    [0.00254, 0.00258], [0.00258, 0.00263], [0.00263, 0.00268],
    [0.00268, 0.00275], [0.00275, 0.00282], [0.00282, 0.00292],
    [0.00292, 0.00304], [0.00304, 0.00321], [0.00321, 0.00345],
    [0.00345, 0.00381], [0.00381, 0.00438], [0.00438, 0.00532],
    [0.00532, 0.00703], [0.00703, 0.01056], [0.01056, 0.01919],
    [0.01919, 0.0477], [0.0477, 0.27727]] ,
35 :
[[0.00107, 0.00231], [0.00231, 0.00236], [0.00236, 0.00239], [0.00239,
    0.00243], [0.00243, 0.00247], [0.00247, 0.00251], [0.00251,
    0.00255], [0.00255, 0.0026], [0.0026, 0.00266], [0.00266, 0.00272],
    [0.00272, 0.00279], [0.00279, 0.00287], [0.00287, 0.00297],
    [0.00297, 0.00308], [0.00308, 0.00322], [0.00322, 0.00338],
    [0.00338, 0.00359], [0.00359, 0.00387], [0.00387, 0.00424],
    [0.00424, 0.00476], [0.00476, 0.00551], [0.00551, 0.00664],
    [0.00664, 0.0085], [0.0085, 0.01174], [0.01174, 0.01789], [0.01789,
    0.03072], [0.03072, 0.05911], [0.05911, 0.1186], [0.1186, 0.21828],
    [0.21828, 0.4535]] ,
36 :
[[0.0, 0.55852], [0.55852, 0.73044], [0.73044, 0.8329], [0.8329,
    0.88287], [0.88287, 0.90438], [0.90438, 0.91174], [0.91174, 0.9149],
     [0.9149, 0.91674], [0.91674, 0.91797], [0.91797, 0.91885],
    [0.91885, 0.9195], [0.9195, 0.92002], [0.92002, 0.92046], [0.92046,
    0.92086], [0.92086, 0.92124], [0.92124, 0.92162], [0.92162, 0.922],
    [0.922, 0.9224], [0.9224, 0.92282], [0.92282, 0.92327], [0.92327,
    0.92375], [0.92375, 0.92429], [0.92429, 0.92491], [0.92491,
    0.92563], [0.92563, 0.9265], [0.9265, 0.9276], [0.9276, 0.92906],
    [0.92906, 0.93117], [0.93117, 0.93501], [0.93501, 0.97777]]
}
```

A2 | Appendix 2 - Probabilities

In this python script, you can find the probabilities corresponding to each interval.

```

right_probs = {
0 :
[2e-05, 0.00019, 0.00014, 0.00025, 0.00011, 0.00016, 0.00016, 0.00019, 0.00019, 8e-05, 0.00011, 0.00011, 0.00011, 2e-05, 0.00016, 8e-05, 8e-05, 0.00014, 8e-05, 0.00019, 0.00014, 8e-05, 5e-05, 0.00014, 2e-05, 0.00019, 5e-05, 0.00014, 5e-05, 0.00056] ,
1 :
[8e-05, 0.00011, 8e-05, 8e-05, 0.00011, 0.00011, 8e-05, 8e-05, 8e-05, 5e-05, 8e-05, 8e-05, 0.00014, 8e-05, 5e-05, 2e-05, 5e-05, 2e-05, 8e-05, 0.00014, 2e-05, 5e-05, 8e-05, 5e-05, 5e-05, 0.00011, 5e-05, 0.00011, 0.00011, 0.00061] ,
2 :
[0.00011, 8e-05, 0.00022, 8e-05, 0.00011, 2e-05, 2e-05, 8e-05, 0.00011, 0.00011, 0.00011, 8e-05, 2e-05, 2e-05, 2e-05, 0.0, 2e-05, 8e-05, 5e-05, 8e-05, 0.00014, 8e-05, 8e-05, 0.0, 8e-05, 5e-05, 8e-05, 8e-05, 8e-05, 0.00053] ,
3 :
[8e-05, 0.00016, 0.00014, 0.00014, 0.00016, 8e-05, 0.00016, 0.00011, 0.00011, 0.00028, 5e-05, 0.00016, 8e-05, 8e-05, 5e-05, 0.00011, 5e-05, 5e-05, 5e-05, 0.0, 8e-05, 0.00022, 2e-05, 2e-05, 2e-05, 2e-05, 0.0, 5e-05, 0.00011, 0.00036] ,
4 :
```

```
[5e-05, 0.00019, 0.00014, 0.00022, 2e-05, 8e-05, 0.00011, 8e-05, 0.00011, 8e-05, 0.00016, 5e-05, 0.0, 5e-05, 5e-05, 0.0, 8e-05, 0.00011, 5e-05, 0.0, 0.0, 0.0, 2e-05, 2e-05, 0.0, 8e-05, 0.0, 0.0, 2e-05, 0.0003] ,
5 :
[2e-05, 0.00011, 0.00011, 5e-05, 5e-05, 0.00016, 0.00011, 5e-05, 5e-05, 0.00014, 2e-05, 2e-05, 8e-05, 2e-05, 0.0, 2e-05, 2e-05, 5e-05, 2e-05, 2e-05, 2e-05, 2e-05, 5e-05, 2e-05, 8e-05, 2e-05, 2e-05, 2e-05, 0.00014, 0.00028] ,
6 :
[0.00011, 0.00014, 0.00011, 8e-05, 0.0, 5e-05, 0.00016, 0.00011, 2e-05, 0.00011, 2e-05, 0.0, 2e-05, 0.0, 0.00014, 5e-05, 0.00011, 2e-05, 2e-05, 2e-05, 2e-05, 0.0, 0.0, 2e-05, 5e-05, 5e-05, 0.0, 2e-05, 0.00033, 0.00028] ,
7 :
[0.00014, 0.00014, 0.00014, 2e-05, 0.00011, 5e-05, 0.00011, 0.0, 0.0, 5e-05, 8e-05, 5e-05, 5e-05, 8e-05, 5e-05, 0.0, 5e-05, 0.0, 8e-05, 0.0, 0.0, 0.0, 2e-05, 0.0, 5e-05, 2e-05, 2e-05, 5e-05, 0.00039, 0.00033] ,
8 :
[8e-05, 0.00014, 8e-05, 0.00014, 8e-05, 5e-05, 5e-05, 0.00011, 2e-05, 2e-05, 0.0, 2e-05, 8e-05, 8e-05, 2e-05, 0.0, 5e-05, 5e-05, 2e-05, 2e-05, 0.0, 2e-05, 2e-05, 5e-05, 0.0, 0.0, 5e-05, 8e-05, 0.00019, 0.00042] ,
9 :
[0.0, 5e-05, 5e-05, 5e-05, 2e-05, 2e-05, 5e-05, 8e-05, 0.00011, 2e-05, 0.00014, 2e-05, 8e-05, 2e-05, 5e-05, 0.0, 2e-05, 0.00011, 2e-05, 0.00011, 0.0, 0.0, 2e-05, 0.0, 5e-05, 2e-05, 8e-05, 5e-05, 0.00014, 0.00042] ,
10 :
[8e-05, 2e-05, 0.00022, 5e-05, 0.00014, 8e-05, 5e-05, 5e-05, 8e-05, 0.0, 8e-05, 8e-05, 2e-05, 8e-05, 0.0, 2e-05, 5e-05, 0.00011, 0.0, 2e-05, 0.0, 0.0, 2e-05, 5e-05, 5e-05, 0.0, 0.00011, 0.00016, 0.00022, 0.00036] ,
11 :
```

```
[0.00014, 2e-05, 0.00019, 5e-05, 0.00011, 2e-05, 0.00011, 0.0, 8e-05, 0.0, 5e-05, 2e-05, 2e-05, 2e-05, 2e-05, 5e-05, 0.0, 5e-05, 5e-05, 5e-05, 0.0, 0.0, 2e-05, 0.0, 2e-05, 0.0, 5e-05, 5e-05, 0.00019, 0.00047] ,
12 :
[2e-05, 0.00011, 8e-05, 0.0, 5e-05, 5e-05, 0.00011, 0.0, 2e-05, 0.0, 5e-05, 5e-05, 2e-05, 5e-05, 2e-05, 2e-05, 2e-05, 2e-05, 2e-05, 0.0, 2e-05, 0.0, 2e-05, 5e-05, 2e-05, 8e-05, 8e-05, 0.00016, 0.00033, 0.00044] ,
13 :
[5e-05, 8e-05, 5e-05, 8e-05, 0.00011, 8e-05, 2e-05, 2e-05, 8e-05, 2e-05, 5e-05, 5e-05, 8e-05, 2e-05, 2e-05, 0.0, 5e-05, 2e-05, 2e-05, 5e-05, 0.0, 5e-05, 0.0, 0.0, 0.0, 0.0, 2e-05, 8e-05, 0.00033, 0.00019] ,
14 :
[0.00011, 8e-05, 2e-05, 5e-05, 0.00011, 2e-05, 5e-05, 0.0, 0.0, 5e-05, 2e-05, 5e-05, 0.0, 5e-05, 0.0, 2e-05, 2e-05, 5e-05, 2e-05, 2e-05, 0.0, 2e-05, 5e-05, 2e-05, 5e-05, 0.0, 5e-05, 0.00016, 0.00011, 0.00036] ,
15 :
[0.00011, 5e-05, 0.00019, 0.00011, 0.00011, 0.0, 0.00011, 0.0, 2e-05, 5e-05, 5e-05, 5e-05, 5e-05, 2e-05, 2e-05, 0.0, 5e-05, 8e-05, 2e-05, 0.0, 2e-05, 0.0, 0.0, 2e-05, 5e-05, 2e-05, 5e-05, 8e-05, 0.00016, 0.00022] ,
16 :
[2e-05, 5e-05, 8e-05, 5e-05, 8e-05, 0.0, 2e-05, 5e-05, 5e-05, 8e-05, 8e-05, 0.0, 8e-05, 5e-05, 2e-05, 0.0, 0.0, 2e-05, 0.0, 5e-05, 0.0, 0.0, 2e-05, 8e-05, 0.0, 0.0, 5e-05, 5e-05, 0.00025, 0.00039] ,
17 :
[5e-05, 0.00016, 2e-05, 0.00011, 2e-05, 2e-05, 0.00014, 0.00011, 8e-05, 5e-05, 0.00011, 0.0, 0.0, 2e-05, 2e-05, 5e-05, 2e-05, 0.0, 8e-05, 0.0, 0.0, 0.0, 2e-05, 2e-05, 0.0, 0.0, 2e-05, 8e-05, 0.00028, 0.00022] ,
18 :
```

```
[5e-05, 5e-05, 2e-05, 5e-05, 2e-05, 0.00011, 2e-05, 2e-05, 8e-05, 2e-05, 2e-05, 0.0, 0.0, 5e-05, 0.0, 2e-05, 0.0, 5e-05, 0.0, 0.0, 5e-05, 2e-05, 0.0, 5e-05, 2e-05, 5e-05, 2e-05, 0.0, 0.00022, 0.00016] ,
19 :
[8e-05, 8e-05, 0.0, 2e-05, 5e-05, 0.00011, 5e-05, 2e-05, 8e-05, 5e-05, 5e-05, 0.0, 0.0, 5e-05, 0.0, 0.0, 2e-05, 2e-05, 0.0, 2e-05, 0.0, 0.0, 0.0, 5e-05, 5e-05, 2e-05, 8e-05, 0.00016, 0.00016, 0.00025] ,
20 :
[5e-05, 2e-05, 5e-05, 2e-05, 2e-05, 0.00016, 0.00011, 8e-05, 0.00014, 5e-05, 0.0, 2e-05, 0.0, 2e-05, 8e-05, 5e-05, 2e-05, 0.0, 2e-05, 0.0, 2e-05, 0.0, 2e-05, 0.0, 8e-05, 0.0, 2e-05, 8e-05, 0.00028, 0.00028] ,
21 :
[0.0, 5e-05, 8e-05, 8e-05, 0.00011, 0.0, 0.0, 2e-05, 5e-05, 5e-05, 0.0, 0.0, 0.0, 2e-05, 2e-05, 0.0, 0.0, 0.0, 5e-05, 2e-05, 5e-05, 0.0, 5e-05, 2e-05, 2e-05, 8e-05, 0.00011, 0.00016, 0.00022, 0.00033] ,
22 :
[2e-05, 0.00019, 0.0, 0.00014, 2e-05, 0.0, 5e-05, 0.00014, 2e-05, 0.0, 2e-05, 2e-05, 2e-05, 2e-05, 5e-05, 0.0, 5e-05, 5e-05, 5e-05, 0.0, 5e-05, 5e-05, 2e-05, 5e-05, 5e-05, 0.00014, 0.0, 0.00016, 0.00033, 0.00047] ,
23 :
[5e-05, 0.00011, 0.00011, 8e-05, 5e-05, 0.00011, 0.00011, 8e-05, 5e-05, 2e-05, 2e-05, 0.0, 0.0, 0.0, 2e-05, 5e-05, 0.0, 2e-05, 2e-05, 0.0, 2e-05, 0.0, 2e-05, 2e-05, 8e-05, 2e-05, 0.00011, 0.00019, 0.00028, 0.00039] ,
24 :
[0.00011, 2e-05, 0.00014, 0.0, 8e-05, 2e-05, 8e-05, 8e-05, 0.0, 2e-05, 8e-05, 2e-05, 0.0, 2e-05, 5e-05, 0.0, 2e-05, 8e-05, 0.0, 0.0, 0.0, 2e-05, 0.00011, 5e-05, 5e-05, 8e-05, 0.00011, 0.00022, 0.00042, 0.00064] ,
25 :
[0.00011, 8e-05, 0.00014, 8e-05, 8e-05, 8e-05, 0.00011, 0.0, 0.00016, 2e-05, 0.0, 0.0, 5e-05, 0.0, 0.0, 8e-05, 0.0, 2e-05, 8e-05, 0.00014,
```

```
    0.0, 8e-05, 5e-05, 2e-05, 0.00011, 5e-05, 8e-05, 0.00033, 0.00044,
    0.00064] ,
26 :
[5e-05, 0.00011, 0.00016, 8e-05, 8e-05, 8e-05, 2e-05, 8e-05, 5e-05,
    0.0, 5e-05, 5e-05, 2e-05, 5e-05, 2e-05, 0.0, 2e-05, 5e-05, 0.0, 8e
    -05, 0.0, 2e-05, 0.0, 0.00014, 2e-05, 2e-05, 0.00014, 0.00025,
    0.00061, 0.00072] ,
27 :
[0.00025, 0.00011, 0.00022, 0.00016, 0.00014, 0.00011, 2e-05, 0.00011,
    8e-05, 2e-05, 8e-05, 0.0, 8e-05, 2e-05, 5e-05, 2e-05, 8e-05, 5e-05,
    2e-05, 0.0, 5e-05, 0.0, 5e-05, 5e-05, 8e-05, 0.00011, 0.00022,
    0.00028, 0.00044, 0.00064] ,
28 :
[0.00014, 0.00016, 0.00011, 8e-05, 0.00014, 5e-05, 0.00019, 0.0,
    0.00011, 8e-05, 0.0, 8e-05, 2e-05, 5e-05, 0.00011, 8e-05, 2e-05,
    0.0, 8e-05, 0.0, 8e-05, 8e-05, 2e-05, 0.00014, 0.00011, 5e-05,
    0.00047, 0.00033, 0.00061, 0.001] ,
29 :
[0.00011, 0.0003, 0.00022, 8e-05, 8e-05, 5e-05, 8e-05, 2e-05, 2e-05, 2e
    -05, 8e-05, 2e-05, 5e-05, 5e-05, 8e-05, 2e-05, 2e-05, 5e-05, 5e-05,
    8e-05, 0.0, 0.00011, 0.00011, 8e-05, 0.00014, 0.00014, 0.00056,
    0.00056, 0.00103, 0.00117] ,
30 :
[0.00014, 0.00019, 0.00011, 5e-05, 0.00014, 2e-05, 5e-05, 0.00014,
    0.00014, 8e-05, 5e-05, 5e-05, 5e-05, 0.0, 2e-05, 5e-05, 0.00011,
    0.00011, 8e-05, 0.00022, 0.00019, 0.00011, 0.00016, 0.00016,
    0.00019, 0.00047, 0.00056, 0.00056, 0.00092, 0.00064] ,
31 :
[0.00025, 0.00022, 8e-05, 0.00019, 0.00014, 8e-05, 0.00011, 5e-05,
    0.00016, 2e-05, 0.00011, 5e-05, 2e-05, 2e-05, 2e-05, 0.00014, 2e-05,
     0.00019, 8e-05, 0.00011, 0.00011, 0.00014, 0.00011, 8e-05, 0.0003,
    0.00053, 0.00084, 0.00086, 0.00092, 0.00072] ,
32 :
[8e-05, 0.00016, 2e-05, 0.00014, 0.00016, 0.00016, 8e-05, 2e-05, 8e-05,
     0.00011, 0.00014, 0.00011, 8e-05, 0.00011, 0.00014, 0.00022, 8e-05,
```

```
    8e-05, 0.00011, 0.00022, 0.00019, 0.00022, 0.00016, 0.00025,
    0.00056, 0.00061, 0.00067, 0.00114, 0.00112, 0.00128] ,
33 :
[8e-05, 8e-05, 5e-05, 0.00011, 5e-05, 0.00011, 0.0, 2e-05, 5e-05,
    0.00016, 0.00014, 0.00014, 0.00014, 8e-05, 5e-05, 8e-05, 8e-05,
    0.00016, 5e-05, 0.00022, 0.00033, 0.00016, 0.00044, 0.00033,
    0.00078, 0.00106, 0.00081, 0.00117, 0.00123, 0.00179] ,
34 :
[8e-05, 8e-05, 8e-05, 0.00011, 0.0, 0.00011, 5e-05, 5e-05, 0.00014,
    0.00022, 8e-05, 0.00014, 8e-05, 0.00016, 0.0003, 0.00022, 0.00025,
    0.00014, 0.00025, 0.00016, 0.00025, 0.00036, 0.00067, 0.0007,
    0.00089, 0.00072, 0.00086, 0.00137, 0.00162, 0.00229] ,
35 :
[2e-05, 2e-05, 0.00011, 0.00016, 0.00025, 0.00016, 0.00014, 0.00016,
    0.00019, 0.00016, 0.0003, 0.00022, 0.00014, 0.00025, 0.00025,
    0.0003, 0.00022, 8e-05, 0.00022, 0.00039, 0.00042, 0.00039, 0.00047,
     0.00067, 0.00123, 0.00137, 0.00137, 0.00193, 0.00238, 0.00263] ,
36 :
[0.54375, 0.63963, 0.70355, 0.76395, 0.82299, 0.90221, 0.93558,
    0.95132, 0.95296, 0.95628, 0.95673, 0.95989, 0.9608, 0.95948,
    0.95916, 0.95949, 0.95805, 0.95656, 0.95664, 0.95586, 0.9562,
    0.95308, 0.95032, 0.94916, 0.94843, 0.94775, 0.94388, 0.94378,
    0.94607, 0.94603]
}
```