Classification of Family Domain of Amino Acid Sequences using CNN – LSTM Architecture

Gokul S – 2018103026

Steven Gilbert – 2018103071

CSE P

Introduction

- In the field of micro biology, understanding the relationship between amino acid sequence and their protein functions can yield far-reaching scientific implications.
- But current lab techniques cannot annotate even 1/3 of microbial protein sequences.
- Predicting the function of a protein from its raw amino acid sequence is a critical step for understanding the relationship between genotype and phenotype.
- We propose a Deep-learning method which consists of a CNN-LSTM hybrid architecture to classify the families of the amino protein sequences.

Pfam

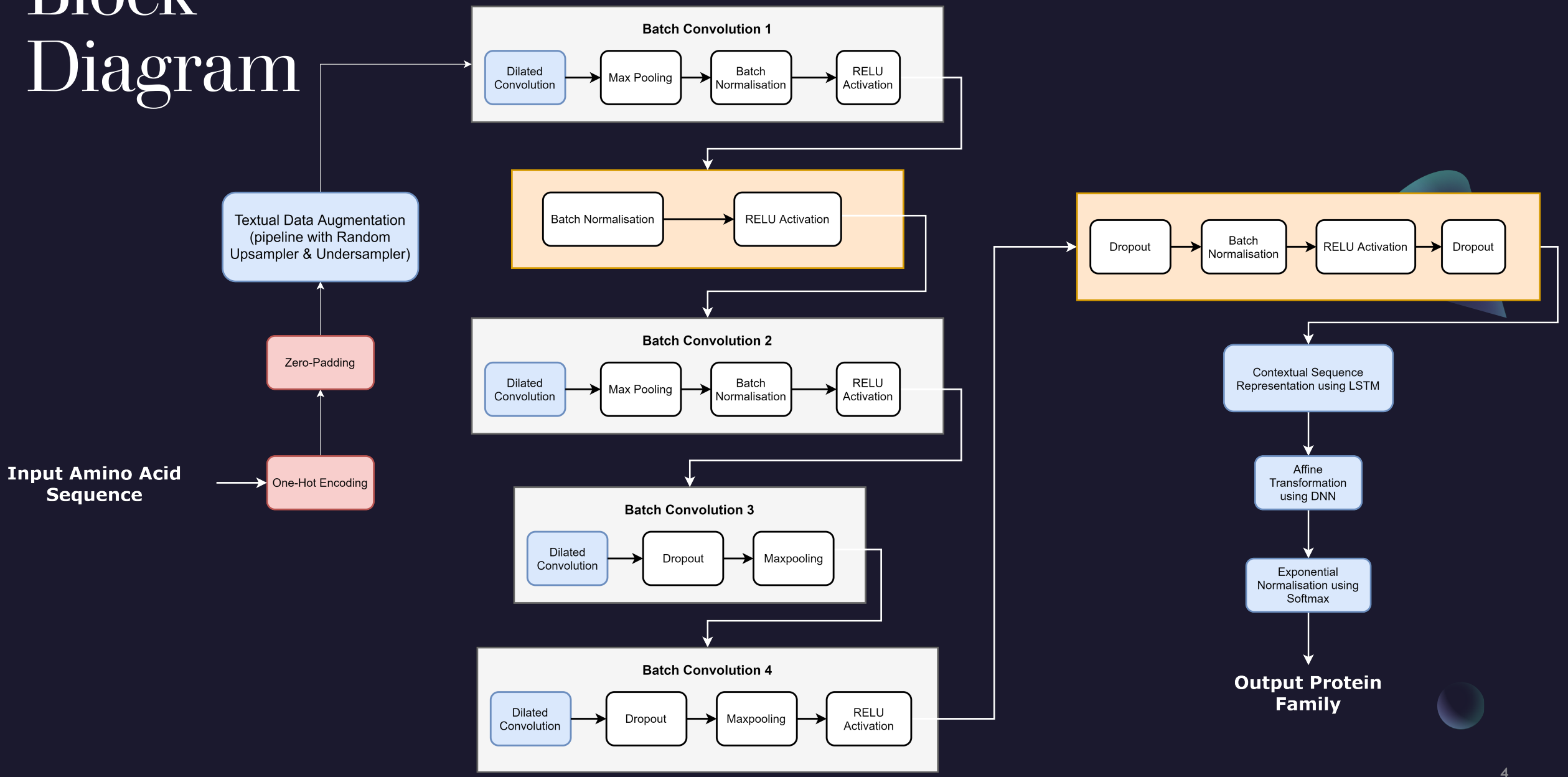
- The Pfam database is a large collection of protein families, each represented by multiple sequence alignments and hidden Markov models(HMM).
- For each family in Pfam :
  - Family Description
  - Multiple alignments
  - Protein domain architecture
  - Species Distribution
  - Known protein structures

```
sequence: HWLQMRDSMNTYNNMVNRCFATCIRSFQEKKVNAEEMDCTKRCVTKFVGYSQRVALRFAE
family_accession: PF02953.15
sequence_name: C5K6N5_PERM5/28-87
aligned_sequence: ....HWLQMRDSMNTYNNMVNRCFATCI..........RS.F....QEKKV
family_id: zf-Tim10_DDP
```

The Dataset contains nearly :

- 1 million sequence examples
- Around 18,000 family output classes

Block Diagram



Methodology

In this module, various plots are used to analyze the structure and distribution of the PFAM dataset in order to check the balance among families

The overall CNN-LSTM architecture is concretely defined here and compiled along with predetermined hyperparameters.

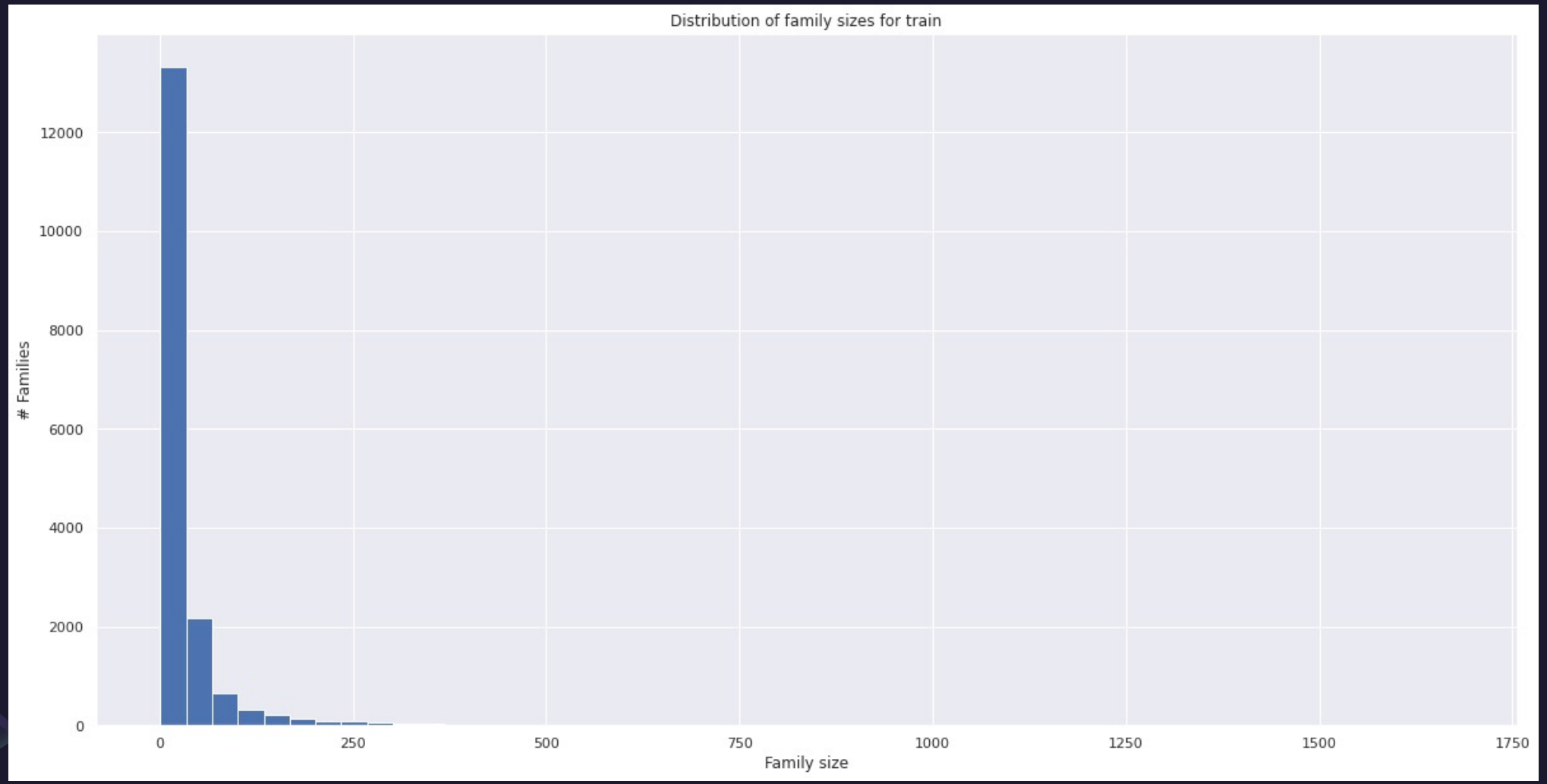
The trained model is loaded to perform Amino acid sequence classification of a completely new and unseen input sequence

Here we first create a dictionary bases on vocabulary. Then encoding occurs with sequence padding along with vectorization

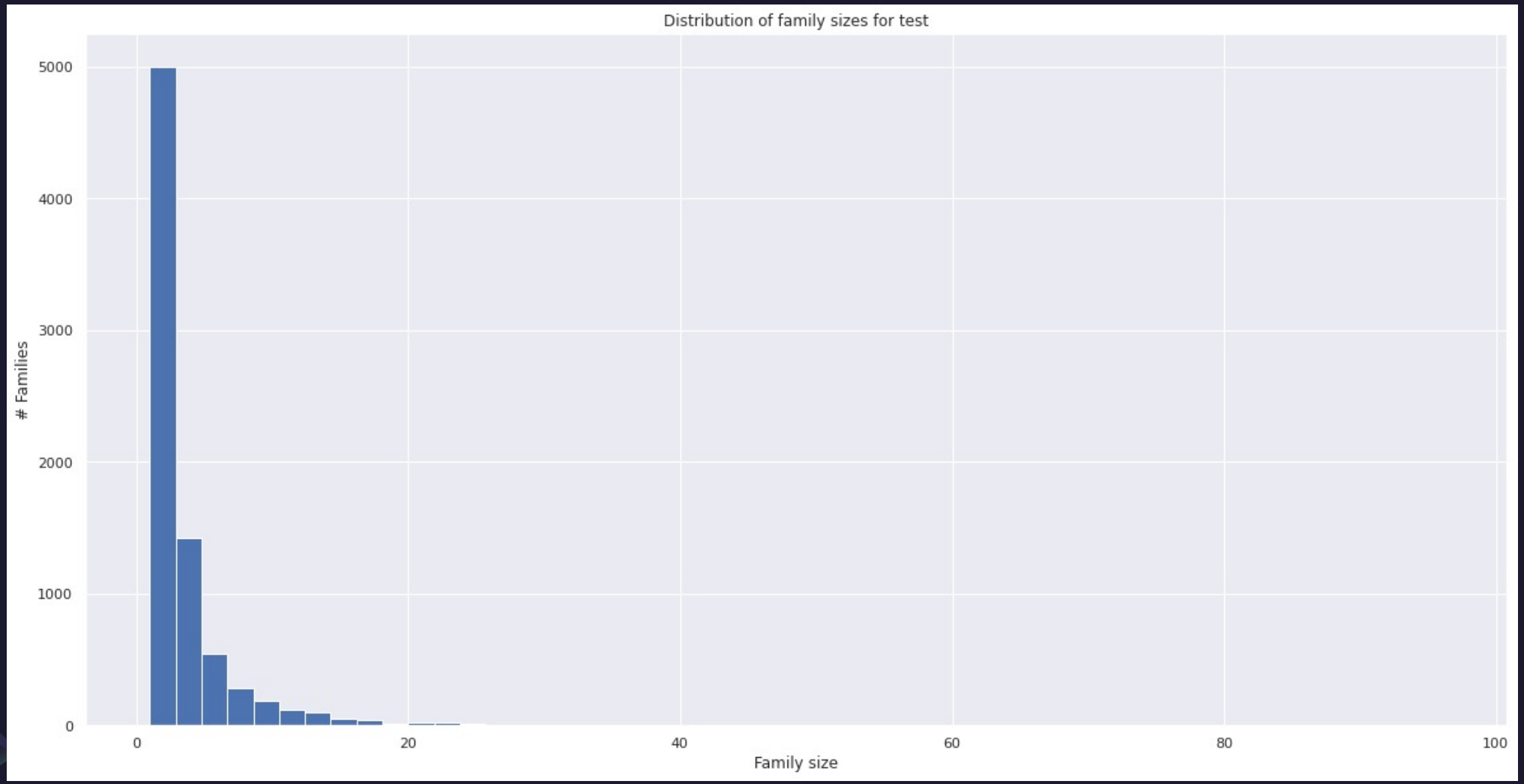
The model is trained on a **Quadro P5000** GPU for 50 epochs to learn the important characteristic features of each of the individual 500,000 sequences.

Exploratory Data Analysis

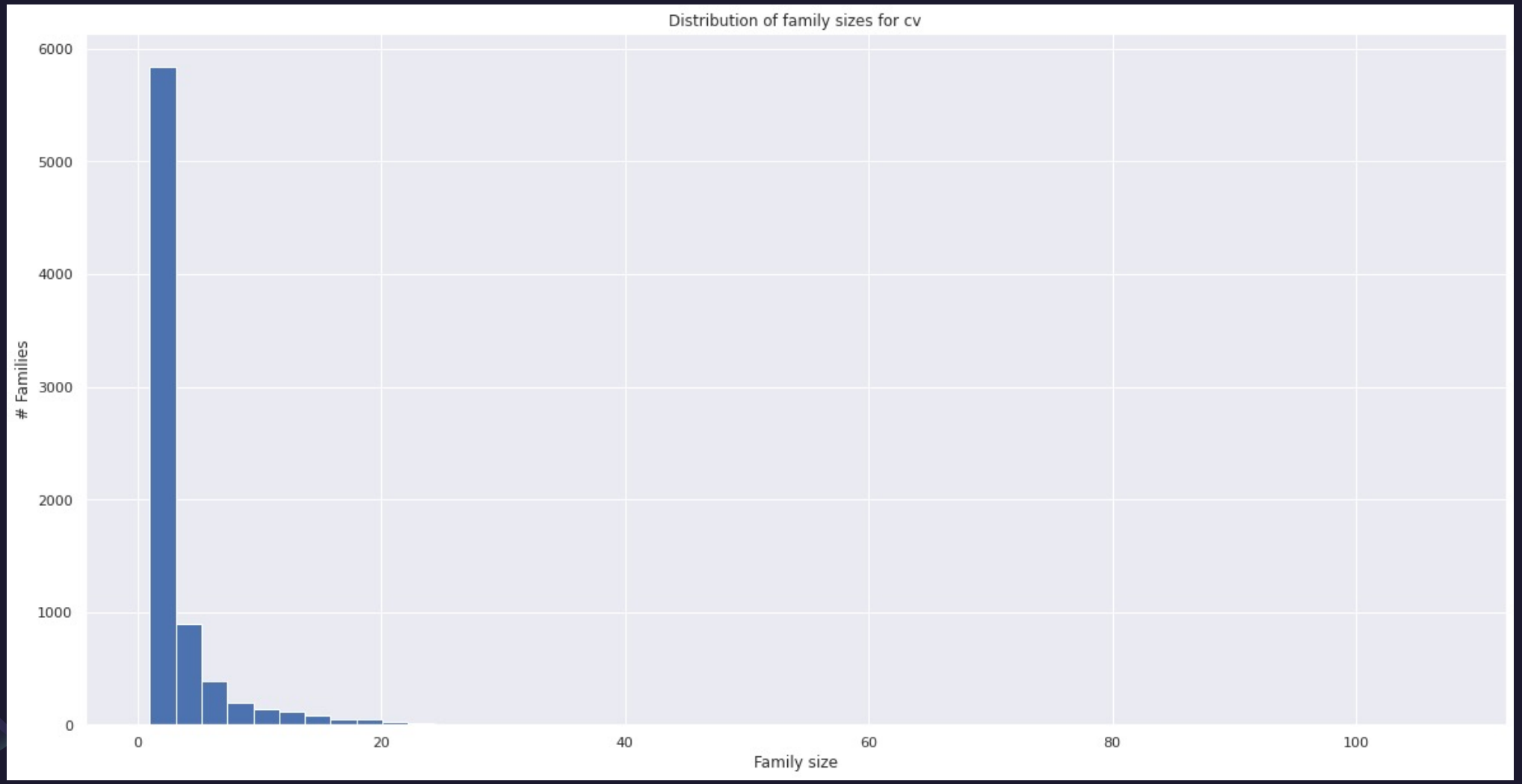
Distribution of Train Family Sizes



Distribution of Test Family Sizes

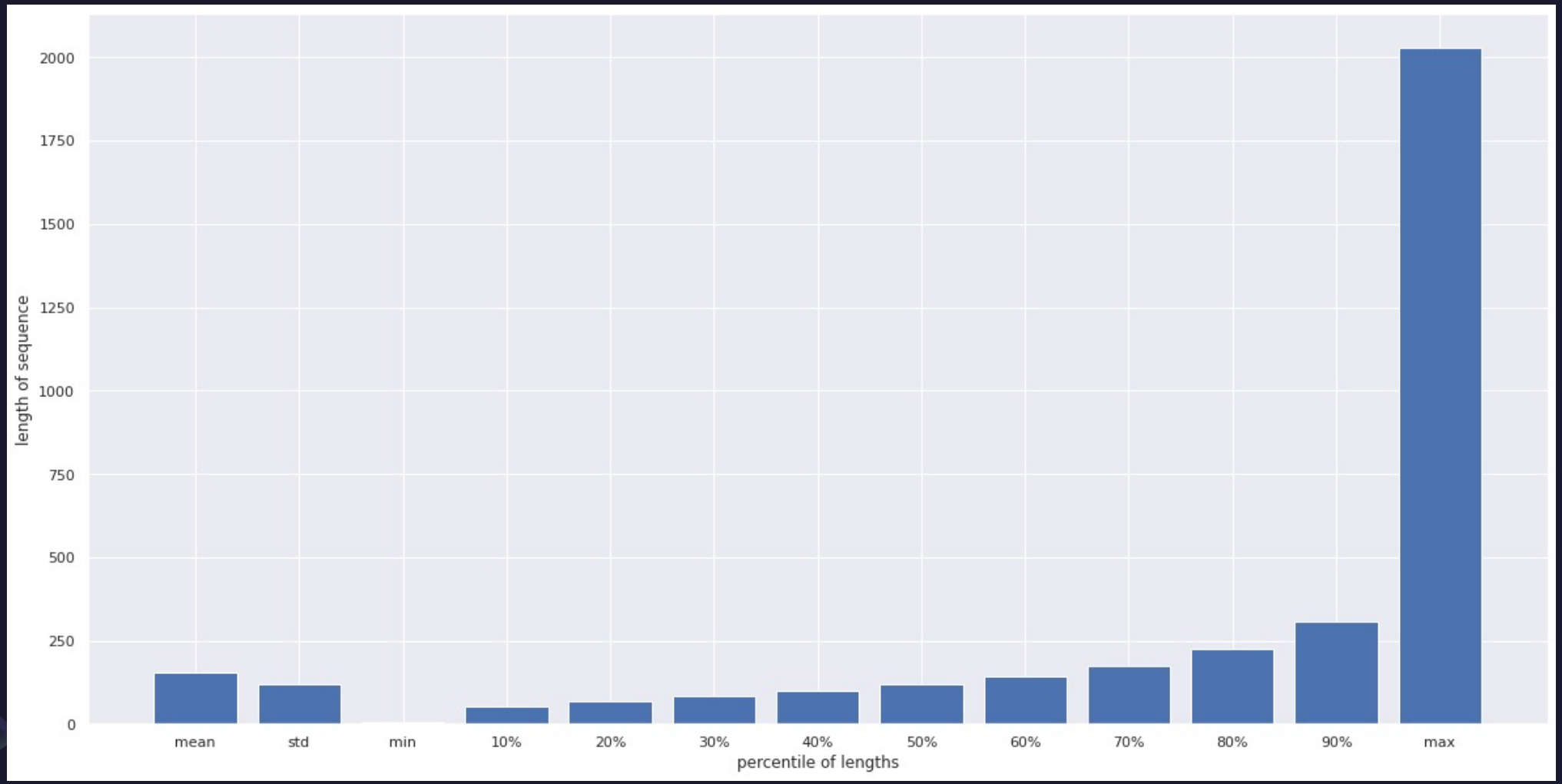


Distribution of Cross Validation (CV) Family Sizes

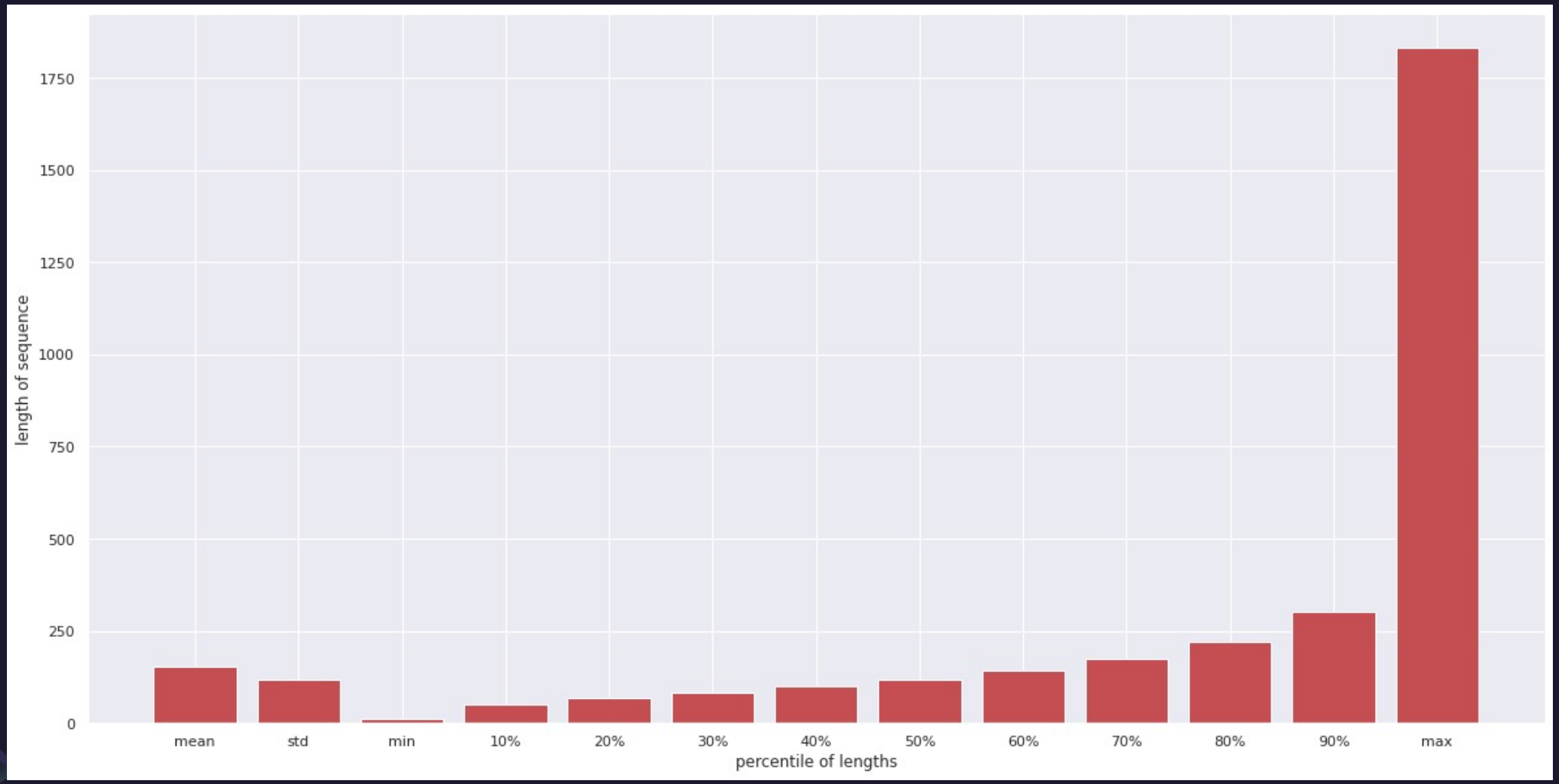


Percentile Distribution of Sequence Length

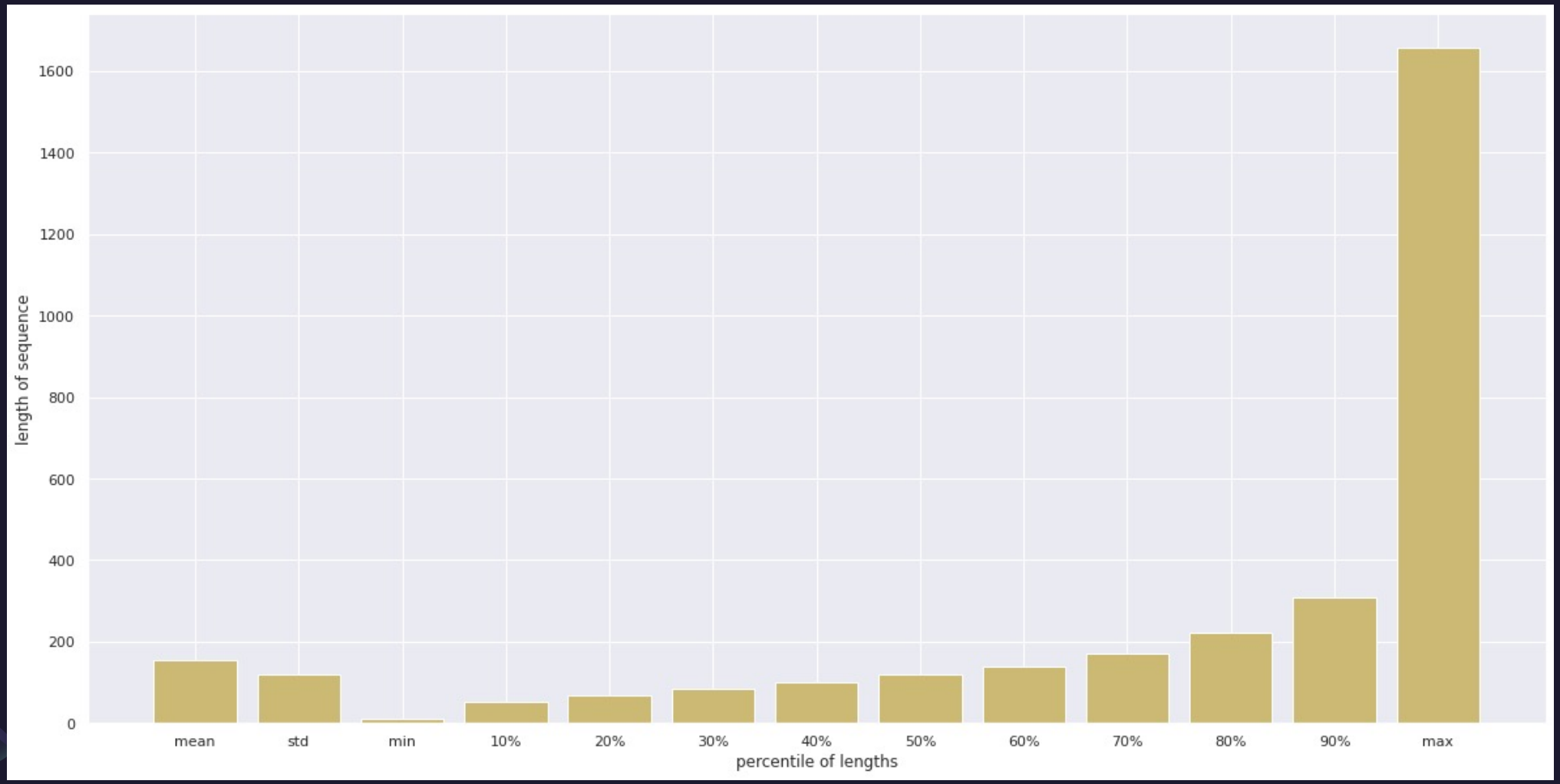
Train



Test

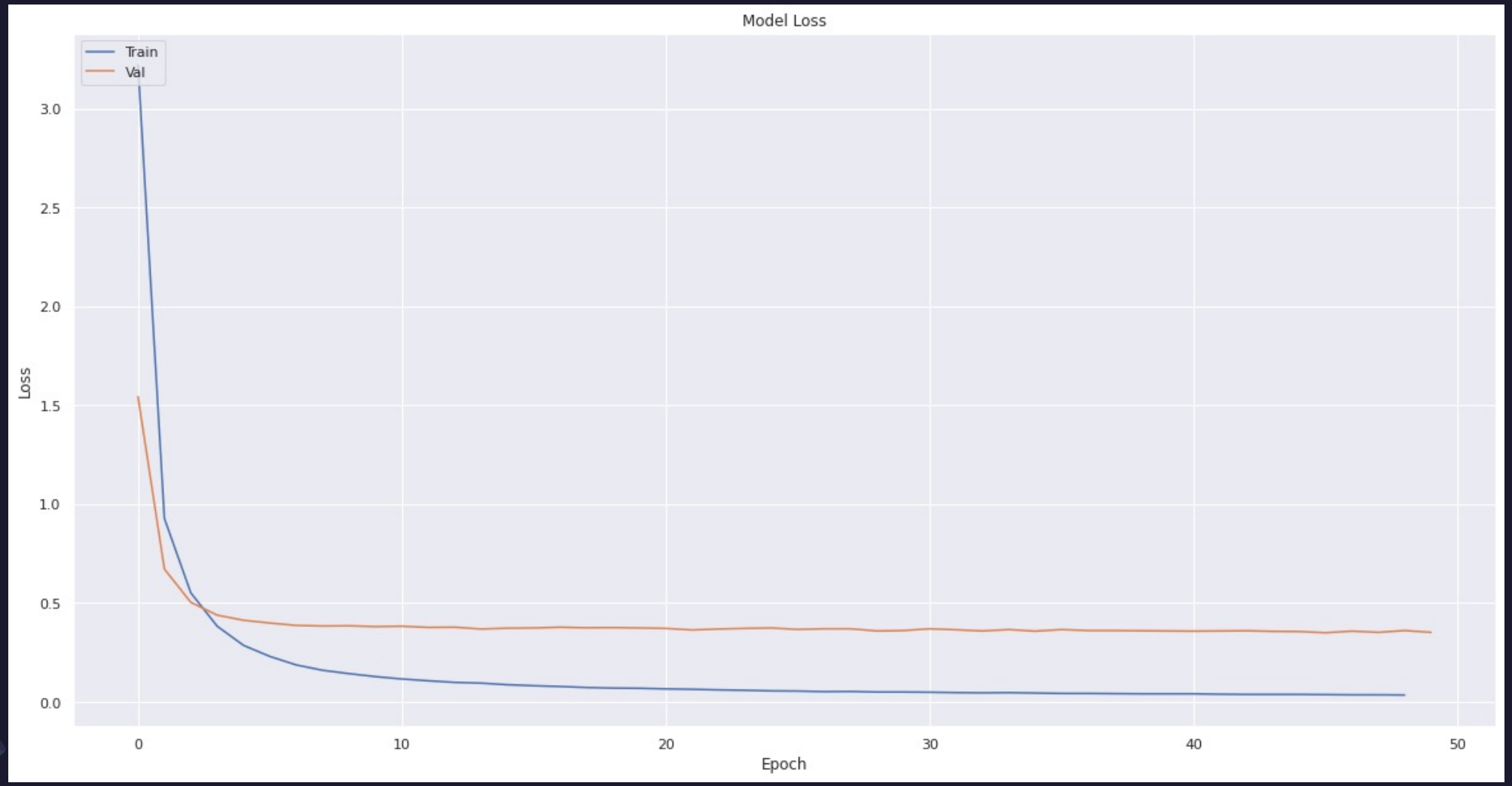

1750
1500
1250
1000
750
500
250
0
length of sequence
mean
std
min
10%
20%
30%
40%
50%
60%
70%
80%
90%
max
percentile of lengths

Cross Validation



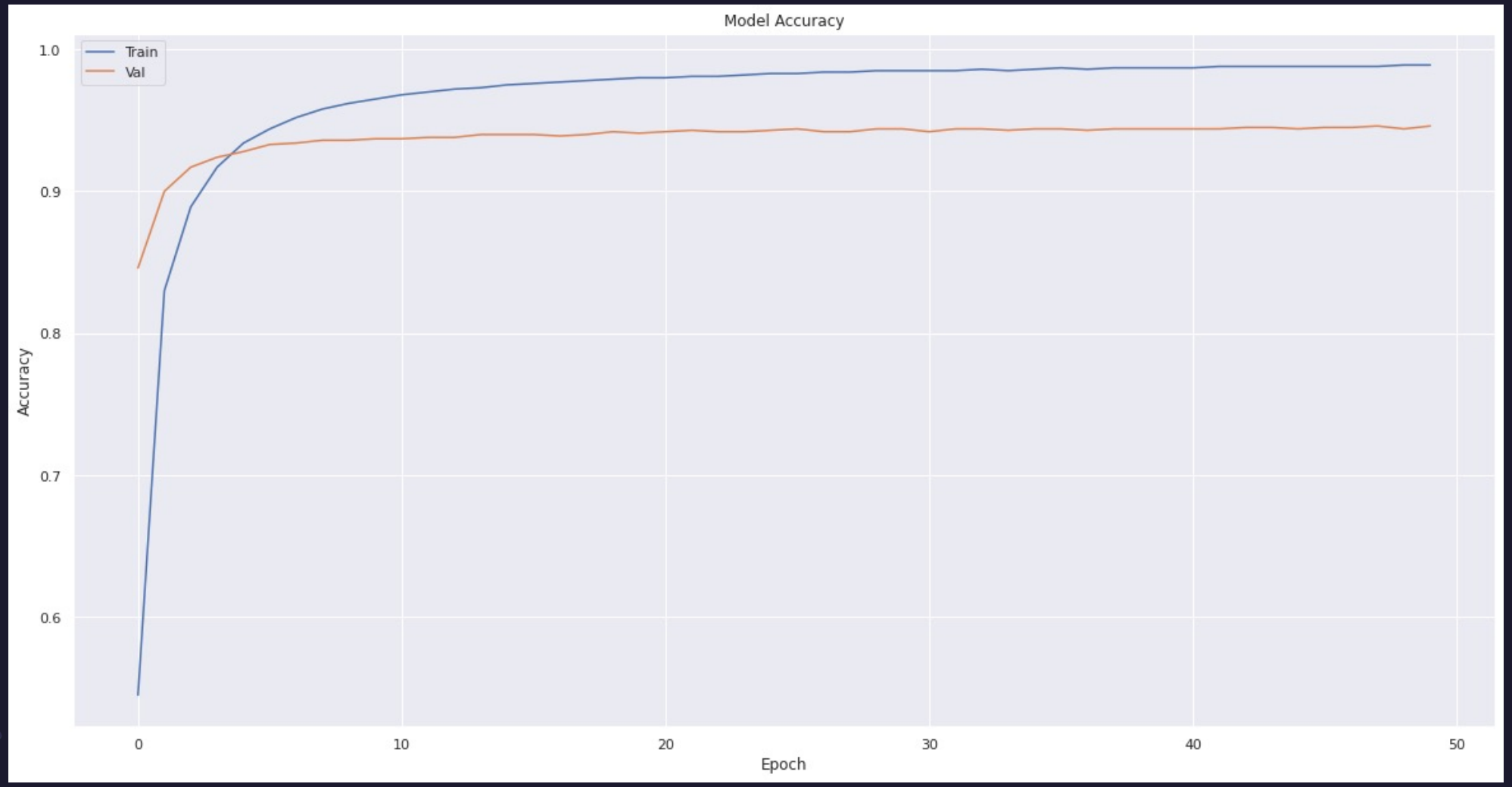
Loss Curve

Model with LSTM and Dilated 1-D convolution



Accuracy Curve

Model with LSTM and Dilated 1-D convolution

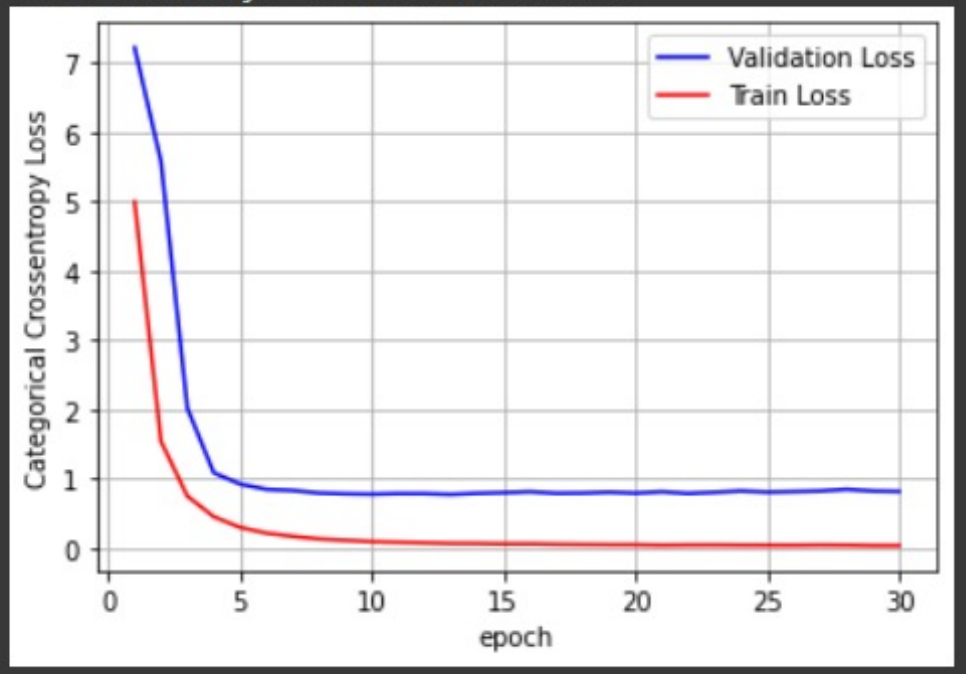


Old Model

Model without LSTM and normal 1-D convolution

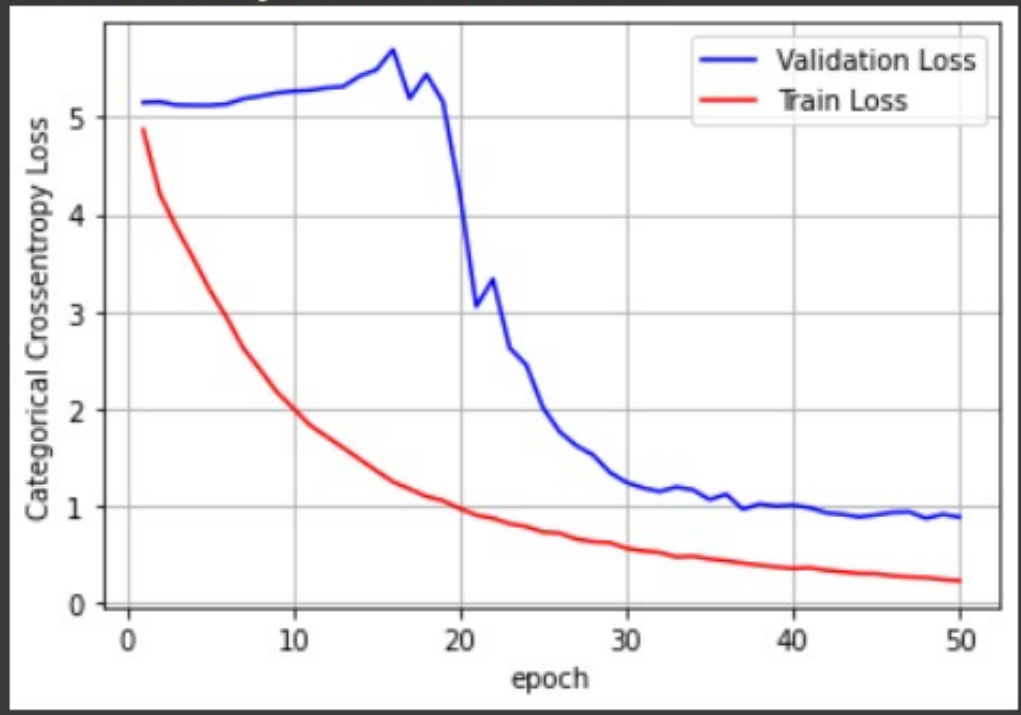
With Data Augmentation

```
Test score: 0.7527622580528259
Test accuracy: 0.8930686116218567
```



Without Data Augmentation

```
Test score: 0.9326633214950562
Test accuracy: 0.732758641242981
```



Performance Metrics

Model with LSTM and Dilated 1-D convolution

```
695/695 [==============================] - 3s 5ms/step - loss: 0.3454 - accuracy: 0.9473
Test loss: 0.3454049229621887
Test accuracy: 0.9473400115966797

+---------------+--------+----------+---------------+
|     Model     | epochs | test loss | test accuracy |
+---------------+--------+----------+---------------+
| Deep CNN-LSTM |   50   |   0.3454  |     0.9473    |
+---------------+--------+----------+---------------+

Classification Report
              precision    recall  f1-score   support

        5809       0.00      0.00      0.00         1
        5810       0.00      0.00      0.00         1
        5811       1.00      1.00      1.00         1
        5812       0.00      0.00      0.00         1
        5813       1.00      1.00      1.00         1
        5814       1.00      1.00      1.00         1
        5815       0.00      0.00      0.00         1
        5816       1.00      1.00      1.00         1
        5817       1.00      1.00      1.00         1
        5818       0.00      0.00      0.00         1
        5819       1.00      1.00      1.00         1
        5820       0.00      0.00      0.00         1
        5821       1.00      1.00      1.00         1
        5822       1.00      1.00      1.00         1
        5823       0.00      0.00      0.00         1
        5824       1.00      1.00      1.00         1
        5825       0.00      0.00      0.00         1
        5826       0.00      0.00      0.00         1
        5827       0.00      0.00      0.00         1
        5828       1.00      1.00      1.00         1
        5829       1.00      1.00      1.00         1
        5830       0.00      0.00      0.00         1
        5831       0.00      0.00      0.00         1

    accuracy                           0.95     22218
   macro avg       0.92      0.91      0.91     22218
weighted avg       0.95      0.95      0.94     22218
```

Predictions

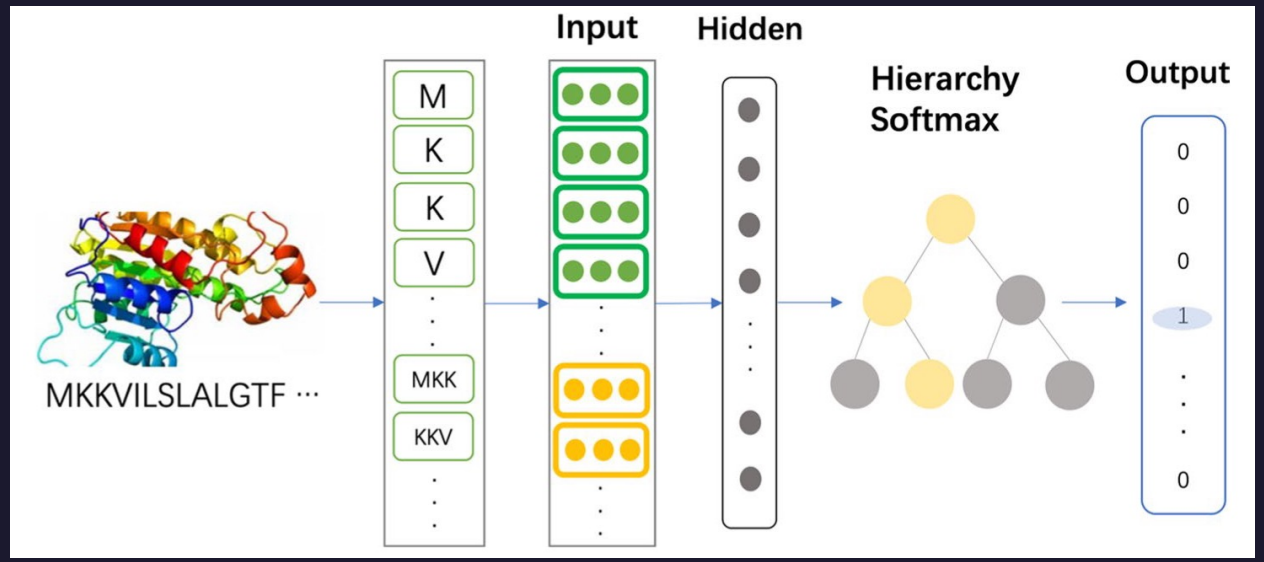
```
def pred(test_seq):
  print("Input Test Sequence: ",test_seq)
  test_seq=[test_seq]
  df2=pd.DataFrame(test_seq)
  df2.columns=['sequence']
  test2  = (df2.sequence).apply(space_in_sequence)
  Test2  = right_padding_with_index(test2[:], max_seq_length)
  ohe_test2   = one_hot(Test2[:])
  y_pred2 = model.predict(ohe_test2)
  c=np.argmax(y_pred2,axis=1)
  print()
  print("Predicted Protein Family for the inputted sequence: ",keys[c])
```

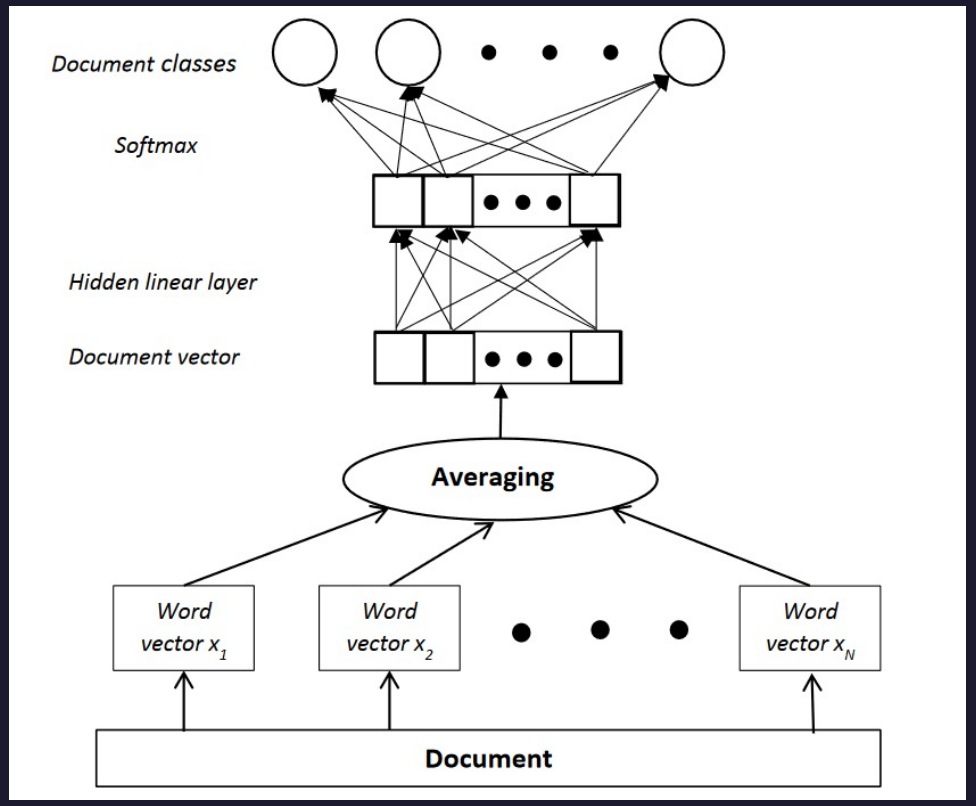
```
[158] pred("HWLQMRDSMNTYNNMVNRCFATCIRSFQEKKVNAEEMDCTKRCVTKFVGYSQRVALRFAE")

Input Test Sequence:  HWLQMRDSMNTYNNMVNRCFATCIRSFQEKKVNAEEMDCTKRCVTKFVGYSQRVALRFAE
100%|██████████| 1/1 [00:00<00:00, 25.50it/s]
Predicted Protein Family for the inputted sequence:  ['PF02953.15']
```

```
[145] def pred(test_seq):
  print("Input Test Sequence: ",test_seq)
  test_seq=[test_seq]
  df2=pd.DataFrame(test_seq)
  df2.columns=['sequence']
  test2  = (df2.sequence).apply(space_in_sequence)
  Test2  = right_padding_with_index(test2[:], max_seq_length)
  ohe_test2   = one_hot(Test2[:])
  y_pred2 = model.predict(ohe_test2)
  c=np.argmax(y_pred2,axis=1)
  print()
  print("Predicted Protein Family for the inputted sequence: ",keys[c])
```

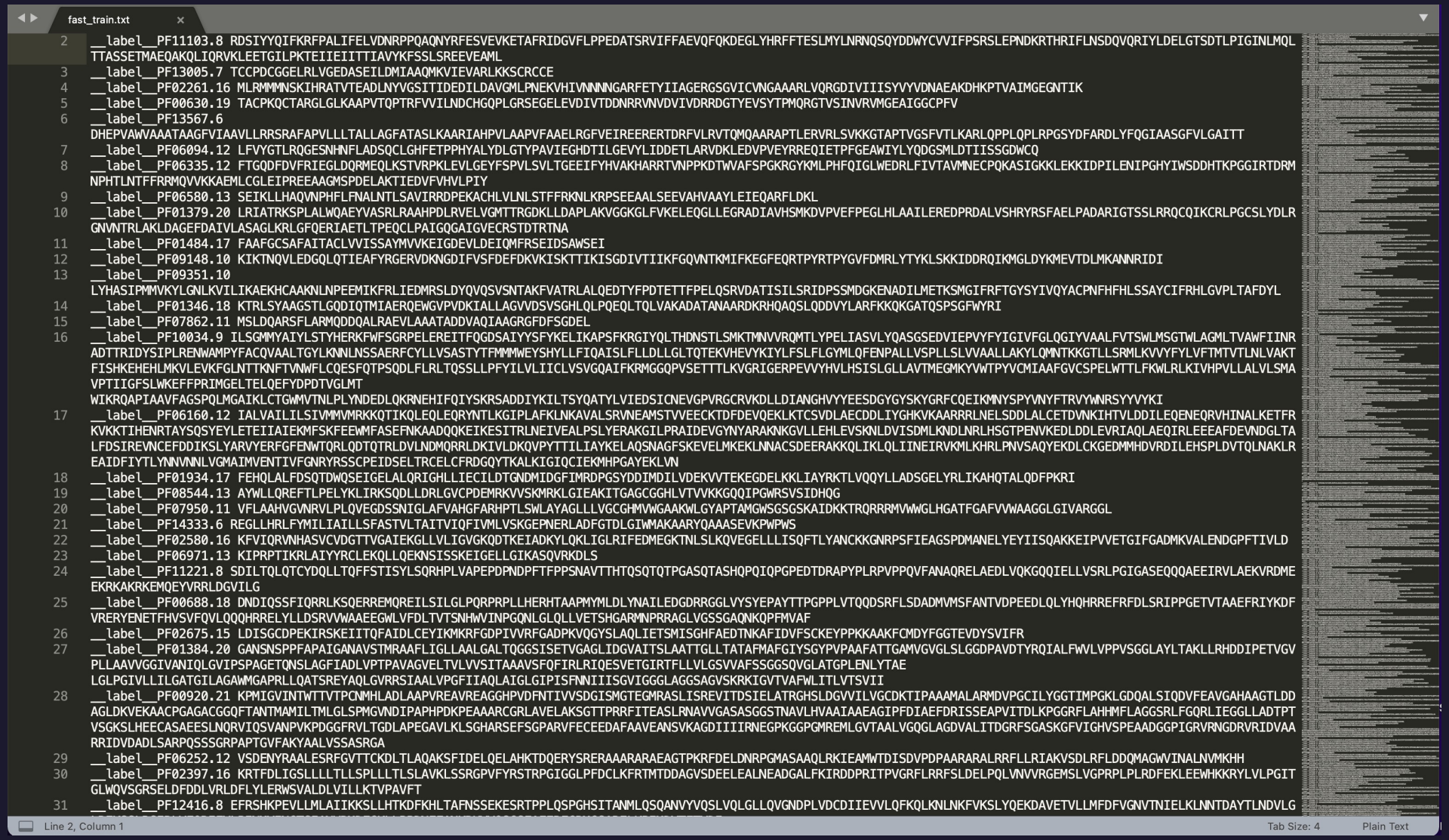
```
[156] pred("GAVRVDVSGGLGTDAMVVSSYLNTDKSLVTVIVNADNQDRDISLAISGGQPAGAVSVYETSAEHDLAPVRNAGADGRLAVKKQSIVTI")

Input Test Sequence:  GAVRVDVSGGLGTDAMVVSSYLNTDKSLVTVIVNADNQDRDISLAISGGQPAGAVSVYETSAEHDLAPVRNAGADGRLAVKKQSIVTI
100%|██████████| 1/1 [00:00<00:00, 19.72it/s]
Predicted Protein Family for the inputted sequence:  ['PF13620.6']
```

FastText – Architecture Diagram

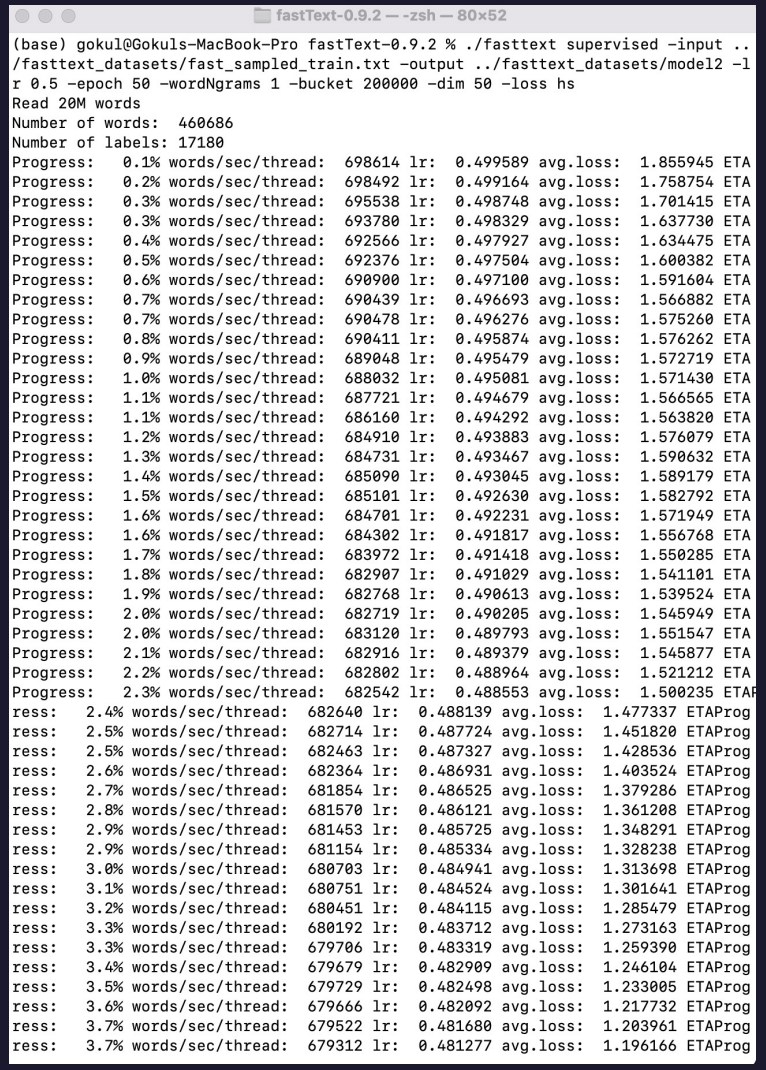


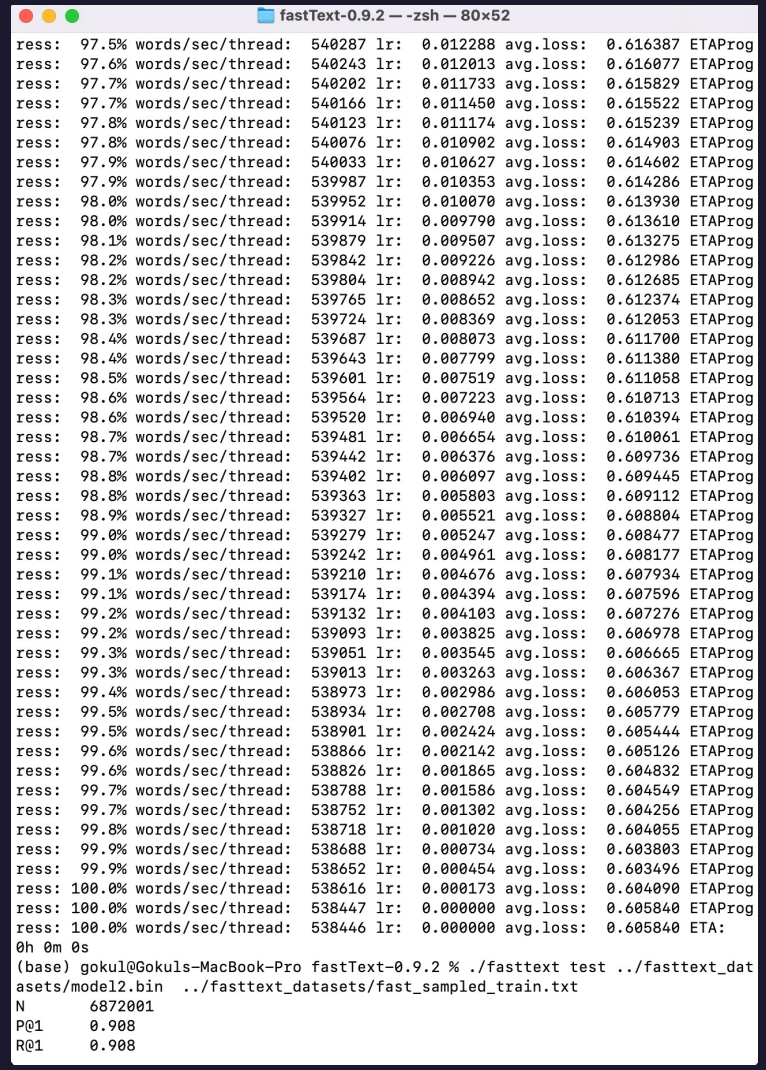


FastText – Input Format



FastText Model – Training Logs





FastText Model

Hyper-Parameters

- FastText Embedding Model = 50
- Loss = Hierarchical Loss
- Bucket Size = 200,000
- Learning Rate = 0.5
- Epochs = 50
- Word N-Grams = 1

Predictions

```
N       6872001
P@1     0.908
R@1     0.908
(base) gokul@Gokuls-MacBook-Pro fastText-0.9.2 %  ./fasttext predict-prob ../fas
ttext_datasets/model2.bin - -1 0.5
STDRVFAAFTGASLVGVARCTRFPEGSLVDGVYVLEEYRHRGFAQRIMRRLIEECGRDGALYLYAKPEHLDFYREMGFEP
__label__PF11992.8 1.00014
VKYQNLVSQLNPHFLFNSLATLESLIYSDRHLAVKFLSQLTRVYRYVLTTRNAKLVSLGEELSFIKDYTDLLQTRFGKGL
__label__PF05315.11 1.00014
KLAGFVQIDDAYLGGERNGGKAGRGSENKQSFLIAVQTDDTFTAPRFVVIEPVRSFDNPSLQDWIARRLAPGCEVYTDGL
ACFRRLEDAGHAHTTLDTSGGRAATEATGARWVNVVLGNLKRAISGVYHAIAQGKYAKRYLAEAAYRFN
__label__PF12762.7 1.00015
PEGYLCHRCHVGGHFIQHCPT
__label__PF13696.6 1.00015
FKVILYGSSIYVVGHVLLSLGAVPFLSYPIRSSLDFSGLFVIAFATGCIKPCVSAFAADQFTEDQKDLRSQFFSFFYFAI
NGGSLFAIIITPILRGRVQCFGNAHCFPLAFGVPGVLMLLALILFLMGWSMYKKHPPSKENVGSKVVAVIYTSLRKMVGG
ASRDKPVTHWLDHAAPEHSQKMIDSTRGLLNVAVIFCPLIFFWALFDQQGSTWVLQARRLDGRVGHFSILPEQIHAINPV
CVLILVPIFEGWVYPALRKITRVTPLRKMAVGGLLTAFSFAIAGVLQLKVNETMEFPPSLGRIYLQRVGNESLISDFRYK
SDGRLIGDGMLPKGRTELDAGIYTFNTGLKNESQEIDISTPNKGYVMAVFRL
__label__PF00854.21 1.00014
^C
(base) gokul@Gokuls-MacBook-Pro fastText-0.9.2 %
```

Inferences

CNN-LSTM Model

- Using a Dilated CNN instead of a normal one proves to be more efficient as the field of reception is spread. This results in a more wider field of view with the same number of kernel parameters and same computational complexity.
- In order to combat the problem of a highly imbalanced dataset, we first over-sample the minority classes using RandomUpsampler and then under-sample the majority classes with an appropriate sampling rate using a RandomUndersampler.
- Using an LSTM layer post 4 sequences of batch convolutions helped the model learn a contextual representation of the amino acid sequence, which is extremely important when it comes to sequence classification problems.
- This improved our test accuracy from 0.8936 to 0.9473 (5.37% improvement).
- Usage of batch normalization sped up training by normalization the hidden layer activation and handling internal covariate shift.

FastText Model

- Under-sampling the majority classes to 400 samples and up-sampling the minority classes to the same size has helped the FastText model to learn a good representation of the PFAM protein database by achieving a P@1 and R@1 score of 0.908.

Summary

Identification of a family of a protein sequence is crucial as it give scientists and researchers a reference point, to which based on the family they can attribute it to known properties. This can prove very useful when identifying the sequence of an unknown disease, to identify its family and to create a potential vaccine based on it's family properties.