

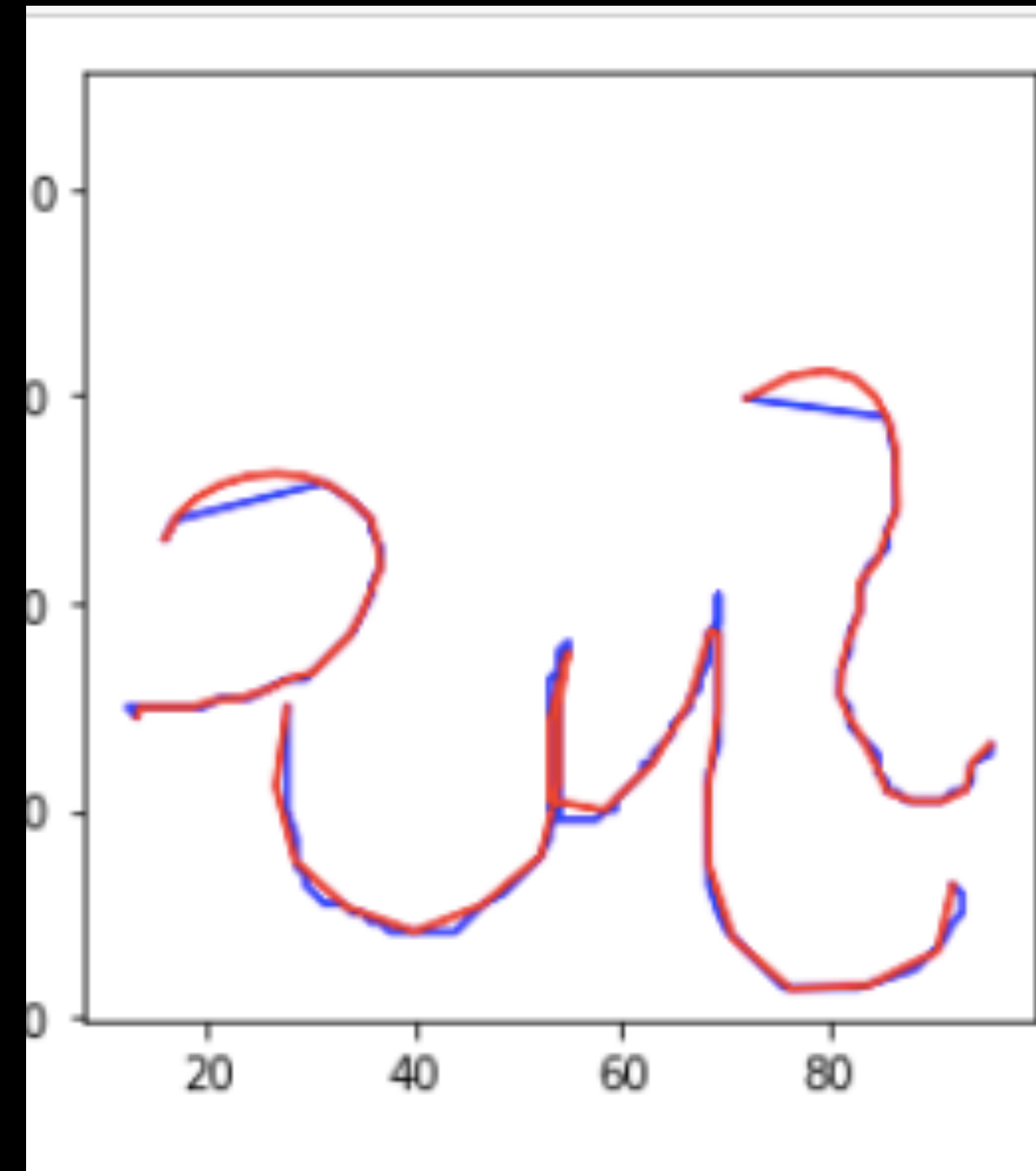
SOM TAMBE (190847)

THE OMNIGLOT CHALLENGE- OWN MODEL

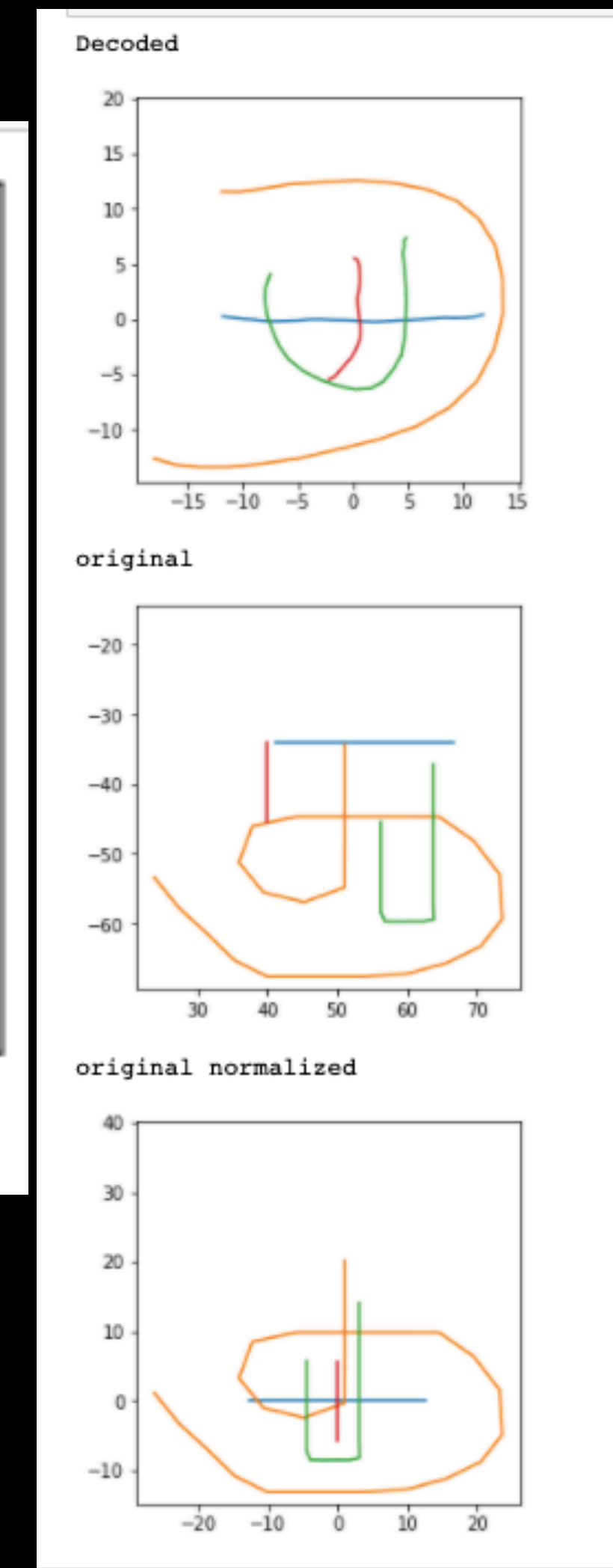


ALGORITHM THAT I FOLLOWED

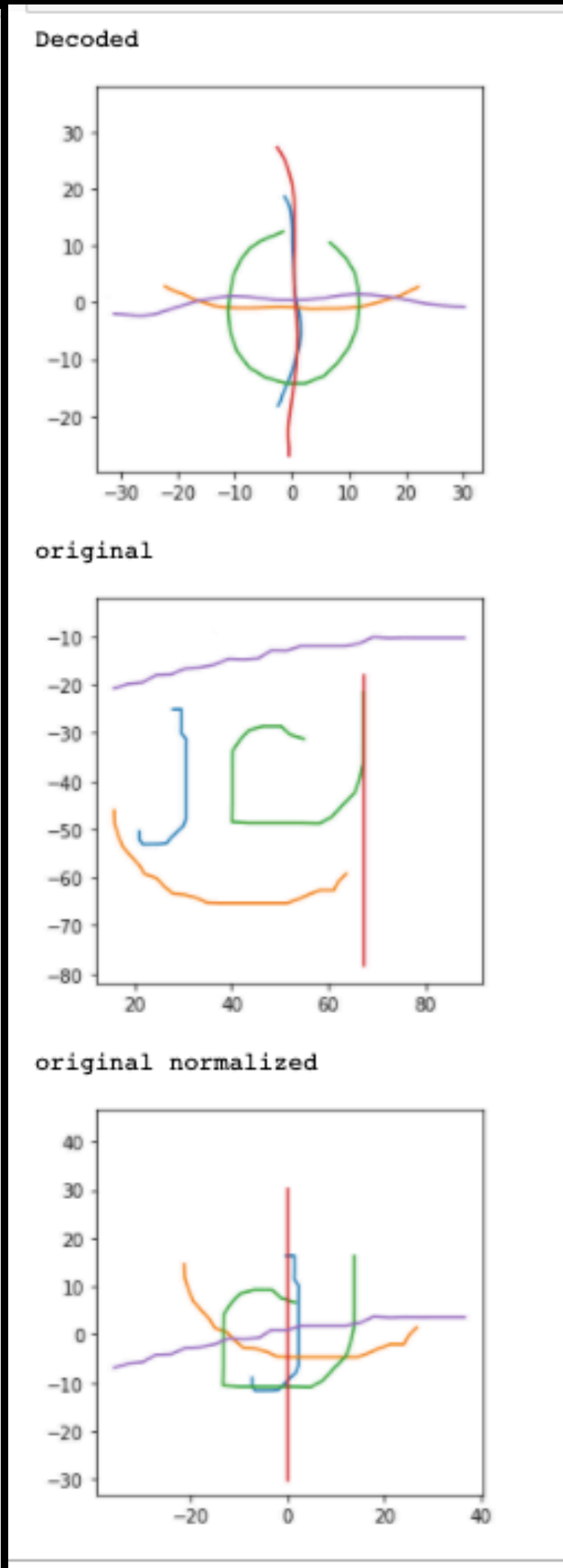
- Convert all strokes (non-constant length, can vary from 1-200) to a 25-point spline (as shown in red in the leftmost figure).
- Use a Variational autoencoder network, feed it this constant 25 length strokes (which were normalized during input, as shown in the right figures), and get a 2 or 3 dimensional latent space representation for the same.
- Total 19280 characters, 48261 strokes in all. Everything was converted to a 25-point spline. Method used was the `scipy.interpolate` function module.



Spline created



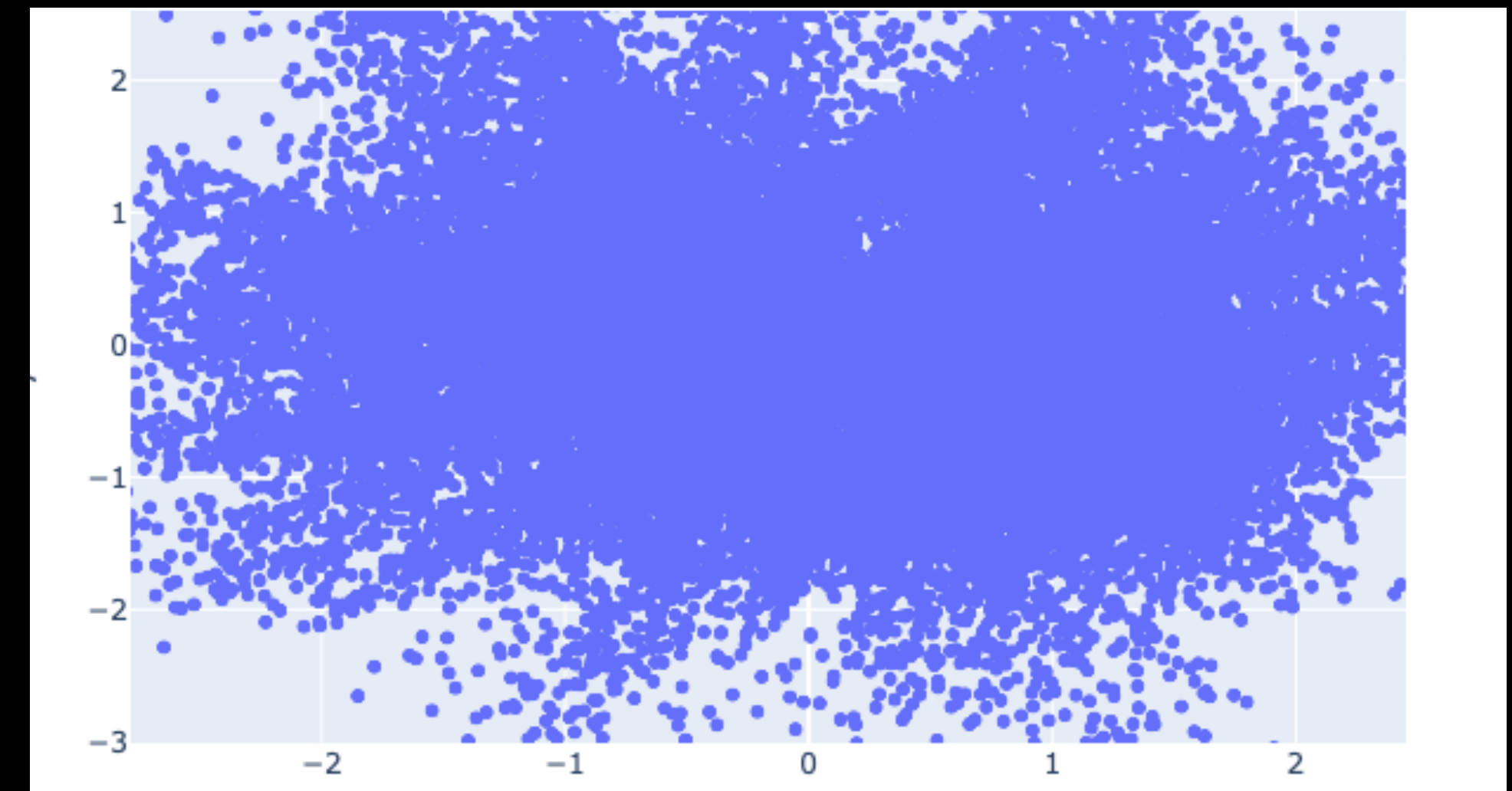
original normalized
was the input



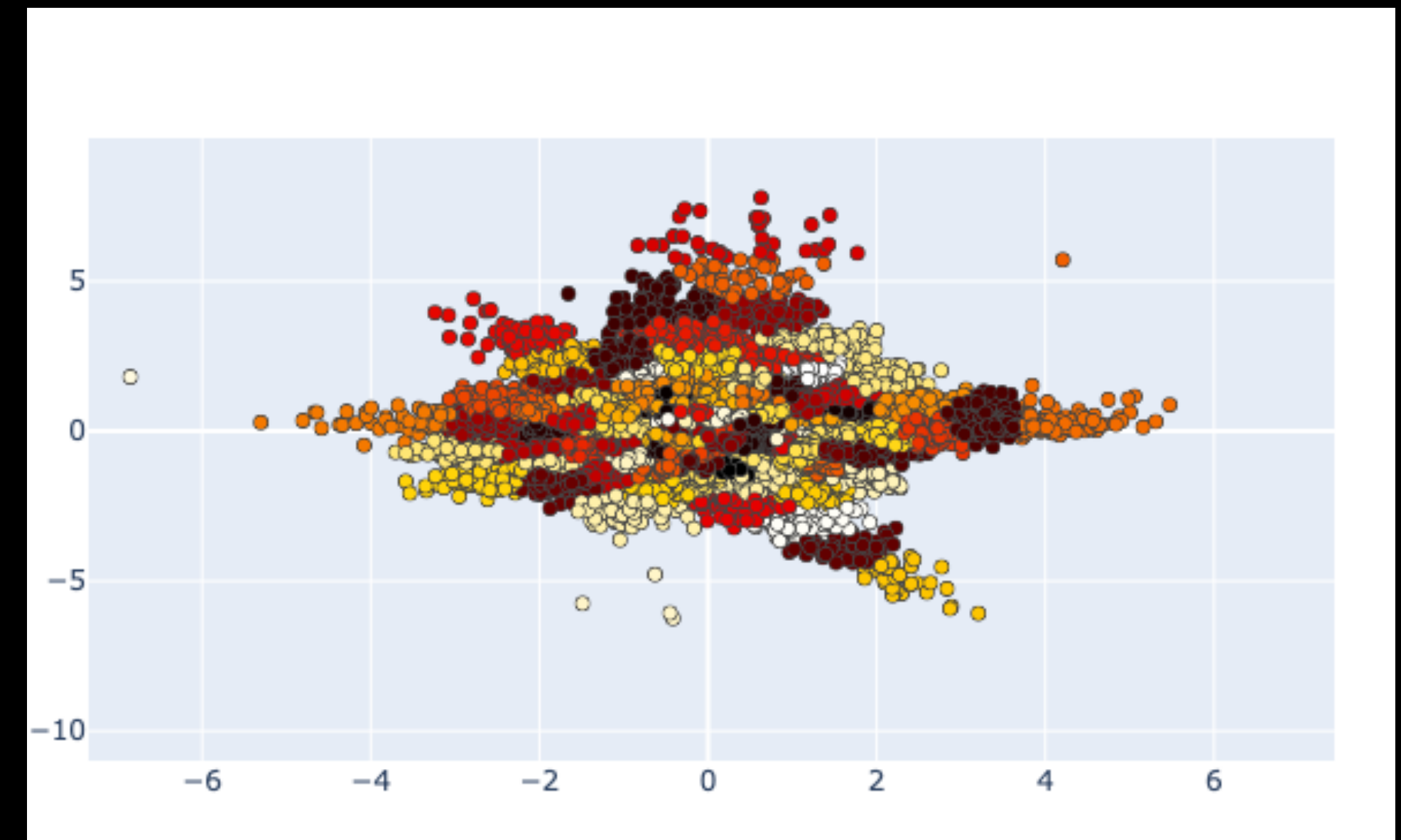
original normalized
was the input

continued-

- After training for 150 epochs, the encoder started giving really good results for 2 as well as 3 dim latent space. 3 dimensional space did not give any visible clusters.
- Plotted the 2 dimensional representation of all strokes taking them as X and Y. Got the above plot (it is little bit zoomed in).
- Then clustered using K-Means clustering into 100 clusters, to get the figure as below.
- Future aims - turn this into one-hot encoded vectors for each and every characters (remember there were 19280 characters, much more strokes), which will denote the strokes which the character contains.
- Then perform supervised learning using these vectors on a convolutional neural network. The (successfully) trained network should give us a network which takes in a character, and gives us a direct representation of its strokes.
- This network can then be used to aid various tasks, such as similarity (when multiple characters are given and you have to recognise similar ones) & classification (when you have to classify a bunch of characters into their classes).



2 dimensional latent space representation



Clustering representation. Different colors represent different clusters.

