**Minutes**

27/10/22

* Looked over work completed this week.
  + Google colab creating a custom dataset class.
  + Discussed pre-processing of data. It’s fine if this takes a little bit of time as it is processing everything before the model runs.
* Dataset loader needs to be looked into this week.
* I can encode amino acids. (e.g A : 10000....)
  + Each amino acid in vector representation.
  + All amino acids equal distances from each other.
* There is a current implementation which takes the window of 15 amino acids as a binary sequence string (e.g ACM is 1000.....010000....000..1000..., where this is 3 vectors stacked horizontally).
  + I could look at representing the amino acids as a matrix – like an image.
    - [ [10000000...], [01000...], [00...010....], …. ].
    - Could use this matrix with a kernel to extract features.
* There are 20 amino acids. We can make an additional ‘buffer amino acid’ for when the window is at the edges of the matrix. So, using 21 amino acids will help us when applying a kernel to find a centre of our matrix section.
  + So now amino acid vectors have length 21. And we can extend the edge by floor(windowSize/2) at both left and right of the matrix.
* Discussed kernels
  + Extracts features.
  + Looked at lecture in week 3.
  + Looked at visual classifier (linked in lecture).
  + Looked at DL book with PyTorch examples embedded throughout.
  + Kernel stride – size of the step it moves by.
* A 21x15 kernel will have a midpoint. This can scan over the matrix and extract features.
* Talked about common 5x5 and 3x3 kernels. If this was used with the protein sequence it would be assuming that some amino acids are more likely to occur next to each other.
  + For example, assuming A, F and G are closely related would determine how we order our vector. To find a relationship between amino acids we would probably have to brute force permutations which would not be totally feasible.
* On ordering of vectors, we will have a hashmap/dictionary to map the amino acid to its vector representation. Because there is not amino acids A-Z.
* Can use a sigmoid as an activation function.
  + This gives us single outputs. This can also be viewed as probability.
  + E.g., amino acid A is predicted with 0.9 and another amino acid is 0.1, or two 0.05 (so the sum of the total probability is 1.0).
  + This lets us classify and predict what amino acid should be here in the sequence.
  + Softmax is similar to sigmoid, however uses multiple values (multiclass problem). This also sums to 1.
* Looked at tutorial for image dataset class.
  + This getitem function loads the images when requested. My dataset isn’t that big so I could possibly pre-process if I want. Or look at making requests when needed.
* Discussed combinatorics of layers and kernels.
  + With deep layers we do not have to trial things exhaustively in a single layer.
  + Example: red truck, blue truck, blue car, red car. In this first layer we can classify car or truck, then in the next layer we classify red or blue. This helps our model learn more unique cases.

Goals for this week:

* Complete the Dataset loader class.
  + This lets us make training batches.
  + Samples for us. Can take sequences and feed them to our model.