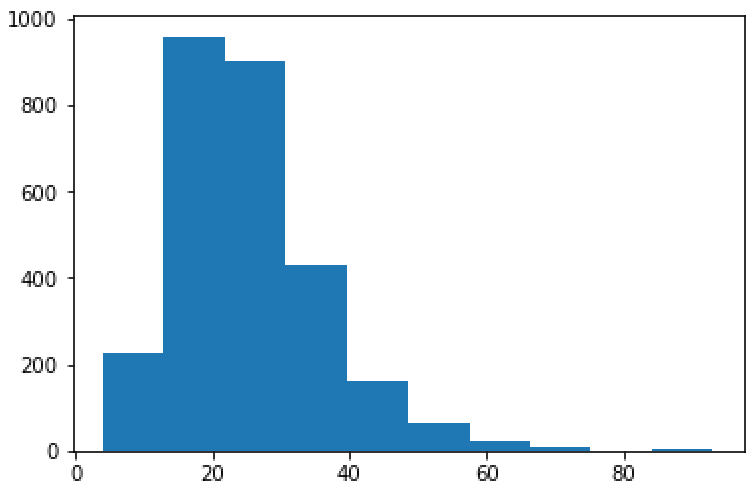
Week 05

Work done

The week has been spent working on the dataset in hopes of obtaining a better accuracy on the test set for our previously implemented network based on the Zhang et al paper.

The main independent variable the researcher studied is the word length of the abstract. We initially assumed that each abstract is 100 words long, but after a statistical analysis we've found that most of the dataset falls somewhere around 30-40 words(mean: 24.98, std: 10.85) . This means that with our initial assumption of 100 words left us with about half the matrix padded with zero vectors. A summary and graph of our dataset can be seen below.



We therefore reduce our dataset to examples up to 48 words. As per our initial architecture, this allows us to concatenate the two proteins we are seeking a relation between on top of the vectorized abstract. If the resulting matrix is under 50 columns, we pad it with zero vectors. In the process we lose 1000 training examples but only a few dozens training examples. This brings our training set to 10000. In theory, this dataset should contain more relevant data. We ran it through our Zhang et al model, for 200 epochs, but results were not encouraging.

Accuracy	Training(200 Epochs)	Test
Zhang	99.247903	58.355319
Zhang+Dropout	97.39207	55.584872
Zhang+Dropout+L2 Regularization	97.61672	65.127528

Since we have concentrated the dataset to the bare essentials, perhaps it's time to take a better look at the embedding. As previously stated, the keras embedding layer based on the 100 dimensional glove vectors leaves out about a third of the vocabulary, and our only option is to train the embedding layer itself. With this in mind, the researcher considered different embeddings for the dataset.

A first consideration would have been fastText. Based on the example provided in the keras documentation, we can see that fastText essentially takes groupings of words(n-grams) and assigns a numerical token to each of them. Then the dataset is augmented by these tokens. In plain English, fastText is a form of data augmentation, but unfortunately it is somewhat stochastic in nature. We cannot determine ahead of time exactly by how many words each abstract will be enriched by, therefore we are unable to make generalizations to dataset. It seems fastText is not what we need.

Instead, we turned to scispacy, and embedding library trained on biomedical, scientific and clinical text. Unlike the glove embedding, scispacy provides a word vector for every word, and has a good chance of a proper encoding for medical terms. The lightweight version of scispacy provides word vectors of 96 dimensions. In keeping with the statistical analysis above, we limit the abstracts to 50 words, and therefore we have a resulting matrix of 50 x 96 after embedding. This superior embedding has its deficits: preprocessing the whole dataset takes a lot longer, up to 20 minutes on the researcher's machine. Still, we have obtained a more numerically complete representation of our dataset, and passed it through our model for 300, and 400 epochs respectively. Results were not encouraging.

Accuracy	Training(300 Epochs)	Test	Training(400 Epochs)	Test
Zhang	88.021493	61.037821	85.930628	58.047491
Zhang+Dropout	82.940888	62.620932	84.748411	55.189097
Zhang+Dropout+L2 Regularization	**79.609185**	**69.613016**	**83.732289**	**61.433595**

The explanation is as follows: if we take a look at our newly compiled model after implementing the scispacy separate embedding, we see that the model parameters have been reduced from 500 000 to 16 000. In essence, this is what the Zhang et al 1D residual convolutional model by itself consists of. In fact, because of the scispacy embedding we see a lot less overfitting, which was likely the result of the glove embedding layer.

Problems

We have done what is possible in preprocessing the limited dataset provided. It is clear that the fault lies in the Zhang et al model. Whether if this is because of a failure in the model itself or the researcher's own implementation it cannot be said. Given the highly technical paper with confounding, similar terms such as filter size vs window size it is a surprise if anyone could replicate the paper at all. In fact, the current implementation is the only one that compiled. This, unfortunately, is not something the researcher has not encountered before, and is endemic in the community. Papers such as these are the inevitable result of the machine learning "culture" of academic elitism that focuses on technical, important sounding papers filled with arcane mathematical formulas instead of practical, intuitive, well explained models that also provide code examples.

To do for next week:

It is clear that we need to abandon the Zhang et al model, which unfortunately puts us right back to where we started. However, the cost of opportunity is rising. Rather than try to improve this obvious limited model that can't crack an accuracy of 70%, perhaps the next week and next sprint would be better spent in searching and implementing alternate models.