**QUEEN'S
UNIVERSITY
BELFAST**

Deep Metric Learning for Proteomics

Lennox, M., Devereux, B., & Robertson, N. (2021). Deep Metric Learning for Proteomics. In M. A. Wani, F. Luo, X. Li, D. Dou, & F. Bonchi (Eds.), *Proceedings - 19th IEEE International Conference on Machine Learning and Applications, ICMLA 2020* (pp. 308-313). [9356221] (Proceedings - 19th IEEE International Conference on Machine Learning and Applications, ICMLA 2020). Institute of Electrical and Electronics Engineers Inc.. <https://doi.org/10.1109/ICMLA51294.2020.00057>

Published in:

Proceedings - 19th IEEE International Conference on Machine Learning and Applications, ICMLA 2020

Document Version:

Peer reviewed version

Queen's University Belfast - Research Portal:

[Link to publication record in Queen's University Belfast Research Portal](#)

Publisher rights

Copyright 2020 IEEE. This work is made available online in accordance with the publisher's policies. Please refer to any applicable terms of use of the publisher.

General rights

Copyright for the publications made accessible via the Queen's University Belfast Research Portal is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

The Research Portal is Queen's institutional repository that provides access to Queen's research output. Every effort has been made to ensure that content in the Research Portal does not infringe any person's rights, or applicable UK laws. If you discover content in the Research Portal that you believe breaches copyright or violates any law, please contact openaccess@qub.ac.uk.

Deep Metric Learning for Proteomics

Abstract—Deep learning has become an innovative tool for predicting the properties of a protein. However, obtaining an accurate predictive model using deep learning methods typically requires a large amount of labelled data, which is expensive and time-consuming to accumulate. Even when optimised, these algorithms are often black boxes, which make it challenging to interpret the decision-making processes that lead to the final prediction. Therefore, there is a demand for innovative modelling techniques that overcome these drawbacks within the space of bioinformatic deep learning. To address these issues, we have designed a modelling scheme that utilises techniques from computer vision. Specifically, we explore how triplet-networks can form a robust model architecture that is capable of learning and ranking proteins from just a few labelled examples. We evaluate our model on a variety of downstream tasks, including peak absorption wavelength, enantioselectivity, plasma membrane localisation, and thermostability. The embedded representations produced by this method show considerable improvement when compared to previous baseline models. Finally, to emphasise that this is an example of white-box deep learning, we visualised the features produced by the algorithm to gain a better understanding as to how the network reaches its prediction for each protein property.

I. INTRODUCTION

Having a limitless supply of labelled data is an ideal scenario for developing any deep learning (DL) model. The sheer volume of examples provided during training enables a DL model to generalise well to the overall task. However, such large labelled datasets do not exist for a majority of cases when modelling proteins [42, 41, 40], which leads to the question of whether deep learning can still be applied in these scenarios. Building a robust deep learning model with a limited amount of labelled examples is not only one of the most challenging aspects for bioinformatics and protein modelling, but for the entire field of deep learning [16, 38, 28]. Issues arise when these complex models begin to overfit to the few examples observed during training, which then produce rather weak predictions when tested on any unseen data. For many years it was believed that the more complex a machine learning algorithm (e.g. deep neural network) was, the more likely it was to overfit. However, recent work in the subfields of natural language processing (NLP) [29, 30, 7], and computer vision [16, 38, 36], have revealed that deep learning models can still be applied with significant effect.

Implementing deep learning models for a task with a limited dataset is by no means a new phenomenon. In particular, many of the most recent advancements within NLP have been based on a technique known as *pre-training*, a method that requires training a deep neural network on a large quantity of unlabelled data before fine-tuning the network on a smaller labelled dataset [14]. The most recent state-of-the-art language models [29, 30, 7] are all based on some form of unsupervised

or self-supervised pre-training (i.e. auto-regressive modelling or auto-encoding) with unlabelled sequence data. Recently, similar approaches have begun to emerge for analysing protein sequences [42, 31, 27]. During pre-training, these networks are tasked with predicting masked out parts of the original input. For auto-regressive deep learning, the masks are sequential, which prevents the network from looking forward when it makes its predictions for each masked token. For auto-encoding, a random portion of the input sequence is masked out. This method does not restrict the network from observing the sequence in any one direction and instead forces the network to include bi-directional context when making its predictions. The utility of pre-training has also been demonstrated in the field of computer vision [6, 43], where the technique of fine-tuning large pre-trained deep neural networks [17, 37, 35] to specific tasks is now standard practise within this field. However, pre-training such large networks takes a great deal of computational time and resources, and therefore the use of such methods are not as common for computational biology.

Other deep learning techniques designed to handle smaller datasets have emerged through the use of metric-based deep learning. These techniques were again popularised through computer vision [39, 15, 25], whereby the network learns a non-linear mapping that can transform a set of images into a feature space before comparing or matching these transformed examples based on a metric (i.e. Euclidean distance, cosine-distance). Popular examples include the use of siamese networks [16], triplet-networks [8], and matching networks [38]. One of the main motivations behind our work is to determine if this form of deep learning is suitable for protein sequence analysis.

Deep neural networks are a prime example of black-box machine learning (i.e. as more layers are included within the model, the more difficult it then becomes to decipher and interpret its decision-making process). This is a significant problem for many fields, including the biological and health sciences, as it is often considered desirable for deep learning algorithms to not only produce state-of-the-art predictions but to also be interpretable for the end-user. One of the benefits of using deep neural networks to model a particular task is that they require little to no feature engineering (i.e. generating task-relevant features from the raw data explicitly). Instead, these networks perform a series of operations before returning a prediction, which can make it difficult to determine how the various parts of the original input are combined to derive important higher-level features (e.g. the relationships between the k-mers within a protein sequence). In the case of modelling protein sequences, it would be beneficial to simulate the

Fig. 1: Triplet loss, based on the Euclidean distance between the anchoring protein and a pair of neighbouring proteins. The positive example should be closer to the anchor than the negative example, with respect to the property of interest (i.e. the values in the circles). The goal of training this model is to encode the information about the triplets and to convert it into a meaningful embedding space.

changes a protein may have when it has been modified, as well as to measure the relative importance of each feature produced by the network. Interpretation of the features extracted by a deep neural network is still an open challenge for the field of machine learning [20, 23, 24]. For deep learning to dominate the modelling landscape of bioinformatics, more attention must be directed towards designing models that better describe the decision-making processes being conducted within these networks.

In this paper, we seek to address these issues through the use of triplet networks, which were previously popularised within the field of image classification [13, 19, 8]. This type of network employs weight sharing to produce encoded representations of the input data (e.g. images), which are clustered based on a triplet loss [13]. To train such a network requires triplets (i.e. anchor, positive and negative) of examples. In the context of protein modelling, it would begin with a single labelled protein sequence that would be used as the anchor. Next, a different pair of labelled protein sequences would form a pair of positive and negative examples, respectively. For the protein to be classified as the positive example, it must possess a label (i.e. measured property) which is closer in absolute value than that of the negative example. Once the triplets have been formed, they are then encoded by the network, which is then optimised via a specific triplet loss that will be discussed in-depth in the following section. Without directly using the corresponding labels to each input, this network is capable of clustering and ranking similar examples together in a semi-supervised fashion. Figure 1 depicts the need for a network to encode each protein in such a way that it reflects the actual measured property in each task (e.g. peak absorption wavelength). Once trained, the encodings produced by the network will reflect the order of the measured property. In this work, we aim to be the first to show the potential of deep

metric learning protein properties through the use of triplet networks. By implementing this style of deep learning, we will be able to use more sophisticated deep neural layers such as convolutional layers to analyse and determine the essential features of each protein sequence without causing the model to overfit to the few labelled examples. Once the network has been suitably trained, we can then encode each protein and analyse how certain modifications to the parent protein can impact its properties.

II. RELATED WORK

Deep neural networks are already a popular option for encoding proteins and extracting useful features from just the primary structure [1, 12, 2]. With the sheer volume of sequencing data being collected and refined by labs across the world, it will be unsurprising to see pre-training becoming a prominent strategy in the field of sequence analysis. With vast resources of known proteins (e.g. the UniProt database [5]), one can now pre-train deep learning algorithms before finally fine-tuning it to a desired area of research, as shown by Rao et al. [31]. This was one of the first significant examples of comparing pre-trained networks to the original baselines. One drawback to the Rao et al. technique was that they only tested their LSTM and transformer networks using a character-based encoding. However, most state-of-the-art language models [29, 30, 7] use a subword encoding algorithm before processing the text (i.e., the text is represented as a sequence of meaningful subword chunks rather than a sequence of characters). The main reason these language models use a subword encoding is to overcome the out-of-vocabulary (OOV) problem. Applied to protein analysis, such an approach has the potential to bolster the original amino-acid vocabulary by efficiently representing potentially functionally relevant sub-sequences. Two of the most prominent subword algorithms within NLP are the byte-pair encoding algorithm (BPE) [10] and the unigram encoding algorithm [18]. The latter of these two compression methods has now become more prevalent in NLP due to it being an entropy-based algorithm, which can produce multiple encoded representations of the original input sequence. By creating a variety of encoded sequences, it enables the model to generalise better to uncommon subsequences and thereby resolves the OOV problem.

Given that a subword algorithm essentially compresses the original input sequence, making it much shorter, a considerable amount of computation time and money can be saved with this simple addition to the analysis of a protein. A subword algorithm can also provide a more extensive vocabulary for analysis at the later stages of the investigation. One of the first examples of using pre-training for a protein sequence was by [42]. Their study consisted of first breaking the proteins into sequences of tri-grams, which were then analysed by a set of Doc2Vec [21] models that used various window sizes. The first drawback to this approach was that they compressed the protein in a rather trivial fashion using the tri-gram encoding method, which is not sensitive to the relative frequency of different tri-grams in the raw sequences. This technique

inevitably leads to a large proportion of poorly represented k-mers. Another issue with this approach is with their choice of embedding model, which adopts a Doc2Vec model to pre-train on the encoded proteins. The model returns one single vector representation for the entire protein, which is problematic when one attempts to investigate specific modifications to a protein or why certain proteins behave differently under various experiments. As an extension to work carried out by Yang et al. with the use of pre-trained models, Lennox et al. tested the use of subword algorithms on protein sequences [22] and found that the Doc2vec models produced better encodings in every downstream task when compared to the tri-gram encoded alternatives [42]. In this investigation, we seek to go beyond both approaches by using a deep neural network to extract the key features of each protein analysed. The network will then provide a rich feature set for each protein as opposed to the single vector representation produced by a Doc2Vec model.

Given the considerable overlap between the techniques used in NLP and computational biology, the approaches used in one field often lend themselves to the other. However, the same cannot be said about computer vision as it can more be complicated to adapt these techniques for computational biology. Consequently, methods such as deep metric learning, which are now a standard in computer vision (e.g. siamese networks [16], triplet-networks [8], and matching networks [38]) have yet to be thoroughly tested within computational biology. The first example of deep metric learning within computer vision was with a siamese style network by [3], which was a non-linear metric learning architecture designed for signature verification. The network was able to learn a set of features that would measure the degree of similarity between two input signatures. Another major implementation of this style of architecture was by Chopra et al. [4], where they used convolutional neural networks in a facial verification task. A convolutional-based siamese network was also used for a dimensionality reduction task [11]. Hoffer et al. noted that these siamese networks are all sensitive to the context of the data [13], as measuring the degree of similarity between all pairs within a dataset are not always feasible, or only a subset of the examples may provide a suitable reference point for making comparisons. Since this investigation is based on modelling a set of regression tasks, a siamese style network would not be beneficial for developing a robust protein embedding space as the labels are continuous as opposed to a set of discrete classes. This would then limit the number of positive pairs (i.e. two proteins that share the same value) one could generate during training, which would hinder the model's performance at generating reliable embeddings for each protein.

The issues surrounding siamese networks were later addressed by Hoffer et al. [13] with another notable adaptation known as a triple network [13]. This three-branched network would take a training triplet (x_a, x_p, x_n) , whereby x_a , x_p , and x_n denote the anchor, positive and negative objects respectively. The network then outputs the following:

$$\begin{aligned} a &= f(x_a) \\ p &= g(x_p) \\ n &= g(x_n) \end{aligned} \quad (1)$$

$$\begin{aligned} D(a, p) &= \|a - p\| \\ D(a, n) &= \|a - n\| \end{aligned} \quad (2)$$

$$L(a, p, n) = \frac{1}{2} \{ \max(0, m + D(a, p) - D(a, n)) \} \quad (3)$$

where the functions f and g denote the branches of the triplet network. For inter-domain learning the anchor branch, f can share weights with the branch g , while for cross-domain learning weights are not shared. An object is termed *positive* if it is similar to the anchor, and vice-versa for the *negative* example. Since the triplet loss is based on minimising the distances between the positive example while also maximising the distances between the negative examples to the anchor, it accounts for the distortion present within the actual embedding space. In the context of computer vision, the triplet network is required to identify whether two images are from the same class as opposed to identifying the class (i.e. the network is weakly supervised). However, in this investigation, we test the triplet network one step further with a set of regression tasks. Instead of identifying each protein and giving it a value, the network must encode a triplet of proteins and rank a pair to the anchor for a particular protein property, as shown in Figure 1.

III. MATERIALS AND METHODS

In a previous analysis on the application of pre-training and subword algorithms on proteins, Lennox et al. concluded that there was a clear benefit to applying such algorithms first to encode a set of proteins [22]. This study revealed that the encoded representations of the proteins produced superior pre-trained encodings when tested on a series of downstream tasks. Both byte-pair and unigram encoding algorithms were analysed and provided the benefit of data compression and an expansion of the original vocabulary. Unlike this previous work, we will not be applying a subword encoding to pre-train our models. Instead, since this is the first example of applying deep metric-learning to this area of research, we will focus on an amino-acid based approach to obtain a stable baseline while implementing a triplet network.

A. Modelling

The triplet network utilised in this investigation is outlined in Figures 2a - 2b. For each task, the proteins are encoded using the deep neural network, as shown in Figure 2a. A trainable embedding layer is applied to transform each amino-acid of the protein into a set of trainable vectors, which are then passed into three separate one-dimensional convolutional layers (128 filters). Each convolutional layer has a different kernel size (i.e. 3, 6 and 9 respectfully) intending to collect important k-mers within the protein sequence. The features

Fig. 2: Overview of the Triplet Network Approach.

produced by each layer are then concatenated with the original embeddings. The final features are then max pooled to form a vector representation for the protein. Each branch of the triplet network 2b begins with the triplet-encoder before being followed up with one final fully-connected layer and L2-normalisation as used in the FaceNet paper [34]. Figure 2b visualises the weight sharing implemented to encode all three proteins during training. Once passed through the network, the three encodings are then evaluated using the triplet-loss.

The loss essentially allows the network to rank the triplet according to the anchoring protein. The objective of the network is then to decide which encoding (i.e. the left or right protein) possesses a measured value that is comparable to the anchoring protein. During training, the network learns to rank a set of proteins based on the 512 max pooled features extracted from the triplet-encoder (Fig. 2a). Once trained, the final encodings produced by the network can be interpreted by a simple regression model to convert the encodings into a set of predicted task values. To remain unbiased in our analysis of this encoding scheme, we apply Gaussian process (GP) regression models [32] based on Matérn kernels with $\nu = 5/2$, to produce the final predicted values for each of the tasks. We choose this regression so that we could make an unbiased comparison of our results to work by Yang et al. [42], and the previous work that implemented subword algorithms [22].

B. Data

In this paper, we cover four downstream tasks, each based on different measured properties. We will now briefly discuss each dataset used in this paper, for more information on the data and the gathering methods implemented in each study, please see citations. The peak absorption wavelength dataset

consists of 80 protein sequences (1–5 mutations), including the original *Gloeobacter violaceus* rhodopsin (GR) parent protein [9]. The enantioselectivity dataset includes 151 protein sequences (1–8 mutations), including the original epoxide hydrolase (EH) parent protein [44]. The plasma membrane localisation dataset consists of 248 protein sequences (1–108 mutations), including the original three-parent proteins [44]. The thermostability (T50) dataset contains 261 protein sequences (1–109 mutations), including the original three-parent proteins [33]. These datasets were chosen for this investigation because they include a diverse range of proteins, and come from libraries constructed from both site-directed mutagenesis and recombination. The diversity across all datasets allow us to thoroughly evaluate the generality of the encoded representations produced by the triplet network.

IV. RESULTS AND DISCUSSION

In this analysis, we evaluated our deep metric learning strategy on a variety of downstream tasks, as shown in Table I. For each task, an eighty-twenty split of the generated triplets from each training dataset was used to train and validate the performance of the model. To highlight, the benefits of using a triplet-network to train our model, we have also included results of a non-triplet based model that was trained from scratch without using triplets. In each of the tasks, the triplet trained version easily surpasses the non-triplet baseline, which indicates that the triplet network is far better at efficiently organising the latent space of each task. We also observed improved mean absolute error (MAE) scores for all four tasks when compared to the original pre-trained baselines [42, 22]. In order to get a better understanding of the encodings produced by the triplet-network, we used t-distributed stochastic neighbour embedding (t-SNE) [26] along with cluster maps,

TABLE I: Results for the four protein downstream tasks.

Model	Encoding	Vocabulary	Absorption	Enantioselectivity	Localization	T50
CNN (Triplet)	Character	20	17.14	5.93	0.63	2.58
CNN (Non-Triplet)	Character	20	25.28	8.01	0.67	3.32
Doc2Vec [22]	Unigram	2000	26.41	6.77	0.65	2.98
		4000	18.09	6.90	0.76	2.80
		8000	20.92	8.58	0.86	2.59
		16000	24.05	7.07	0.77	3.33
		32000	21.98	9.53	0.76	2.96
Doc2Vec [22]	BPE	2000	23.83	10.38	0.66	2.70
		4000	20.80	9.76	0.67	3.01
		8000	18.46	6.72	0.75	2.75
		16000	20.64	6.08	0.73	2.76
		32000	24.27	7.03	0.67	2.80
Doc2Vec [42]	Tri-gram	8000	23.30	9.14	0.73	2.91
Doc2Vec [22]	Character	20	46.08	12.55	0.81	4.32

Notes: Mean Absolute Error (MAE) between the actual test values and the predicted test values.

Fig. 3: T-SNE and cluster plots, depicting the relationship between the similarity structure of the learned embedding space and the protein properties of interest (see text for details).

Fig. 4: Saliency maps highlighting the importance of each amino-acid for three proteins taken from the peak absorption task. The original parent protein is outlined in green, the least and most absorbent mutated versions of the parent protein are outlined in blue and red respectively. The mutations can be identified via the red lettering.

as shown in figure 3 (with all t-SNE projections using a perplexity of 30). The combination of both plots provides us with a clear visualisation of how the network perceives each protein within the individual tasks.

In Figure 3, the triplet network produces a set of encodings that can easily be clustered based on peak absorption wavelength, with the parent protein appearing central as expected, given its recorded value. From these figures, one can see the correlation between the number of mutations a protein has and its absorption value. We can also discern how the network quickly identifies the minor mutations to the original protein. These plots then allow us to observe the contribution each mutation has to the final feature vector. Visualising the encodings in this fashion allows one to identify the mutations that complement the various parts of the original parent protein.

Similarly to the absorption task, we saw an improvement in the enantioselectivity task using this encoding scheme, as shown in Table I. The t-SNE and cluster maps of this task, figure 3, show how the triplet encodings link the information with regards to mutation to the actual property of enantioselectivity. Again, the network is capable of successfully ranking the proteins as these plots indicate the opposite effect of absorption as the proteins with fewer mutations tending to have a higher value for enantioselectivity.

The results for the plasma membrane localisation task fared similarly to both the absorption and the enantioselectivity tasks. The triplet encodings once again improved upon the

original baselines shown in Table I. When considering the t-SNE and correlation plots in Figure 3, we can see that the fewer mutations the protein has, the more likely it is to have a high localisation value, which is similar to that of the enantioselectivity task. By visualising the encodings produced by the triplet network, we can determine that one of the three-parent proteins is more likely to possess a higher localisation value. The more mutations a protein has, the more central it is in the t-SNE plot, and the less likely it is to have a high value.

The last of the four protein downstream tasks was based on modelling thermostability (i.e. T50) values. This task again benefited from the use of the triplet network, with clear improvements over the original baselines outlined in Table I. This result is unsurprising given the results of the other tasks. When we visualise the encodings produced by the triplet network on Figure 3, we can see how the network was able to cluster the proteins with high T50 values away from those with the lower values. From these plots, it should be noted that the network can recognise the three individual parent proteins within this task, as is evident in their separation within the t-SNE plots. Unlike in the previous three tasks tested, the network seems to cluster the protein sequences solely on the actual property of interest. Therefore, not basing the prediction on the frequency of mutations, but instead focusing on the actual mutations present within the protein. This is apparent given the large cluster of proteins sharing a high T50 value. We can also recognise that the network has cluster-specific

parent proteins closer to the high T50 value cluster, indicating which of the three original parent proteins have more potential for the task.

To provide a more in-depth insight into the essential features produced by our model, we will focus on a single example based on the peak absorption task. In Figure 4, we have included a set of saliency maps with the original parent protein from the task is outlined in green. In this figure, we have highlighted the importance of each amino-acid within the protein sequence based on the features present in the final max-pooled representation. Below the parent protein, we have also included two mutated versions of the parent protein. The least absorbent mutated version of the parent protein is outlined in the blue, and the most absorbent is outlined in red respectfully. When comparing all three proteins in Figure 4, one can see that the mutations (red lettering) align with what is model deems essential from the original parent protein. This figure illuminates the model’s ability to capture known mechanistic impacts of specific mutations without any prior knowledge about the biochemistry.

V. CONCLUSION

This investigation has shown how one can still utilise a deep neural network to produce state-of-art results even when there is a limited amount of labelled data. This modelling scheme provides an insight into how the neural network forms each prediction as one can visualise the impact of modifying the protein. In doing so, we can gain a more precise look as to why specific proteins behave differently. When visualising the embeddings of all four downstream tasks, we compared the correlations of each protein to their corresponding measured values of interest and the number of mutations the protein compared with the parent. In all examples, we were able to see how the encodings generated contained information at both levels. From the results, the triplet network appeared to produce more detailed robust encodings with regards to the measured properties when compared to the non-triplet and previous Doc2Vec pre-trained baselines. Although the cluster maps did indicate some clustering within the data, there is still room for improvement with regards to separation between the individual proteins. These results also introduce the question of whether some other metric-based training scheme could further improve the performance, such as a matching network. Additionally, there may be room for improvement with the use of a recurrent layer (i.e. long-short term memory network or transformer network). This will allow the network to model the long-range dependencies within the protein.

In this paper, we have shown that applying modern deep learning techniques to encode a protein can improve upon previous baselines and enhance the amount of information gleaned from each task. The triplet style network allowed us to take full advantage of a limited dataset and displayed the potential of deep learning in this problem domain when suitable techniques are applied.

REFERENCES

- [1] Babak Alipanahi et al. “Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning”. In: *Nature biotechnology* 33.8 (2015), pp. 831–838.
- [2] José Juan Almagro Armenteros et al. “DeepLoc: prediction of protein subcellular localization using deep learning”. In: *Bioinformatics* 33.21 (2017), pp. 3387–3395.
- [3] Jane Bromley et al. “Signature verification using a” siamese” time delay neural network”. In: *Advances in neural information processing systems*. 1994, pp. 737–744.
- [4] Sumit Chopra, Raia Hadsell, and Yann LeCun. “Learning a similarity metric discriminatively, with application to face verification”. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*. Vol. 1. IEEE. 2005, pp. 539–546.
- [5] UniProt Consortium. “UniProt: a hub for protein information”. In: *Nucleic acids research* 43.D1 (2014), pp. D204–D212.
- [6] Jia Deng et al. “Imagenet: A large-scale hierarchical image database”. In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.
- [7] Jacob Devlin et al. “Bert: Pre-training of deep bidirectional transformers for language understanding”. In: *arXiv preprint arXiv:1810.04805* (2018).
- [8] Xingping Dong and Jianbing Shen. “Triplet loss in siamese network for object tracking”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 459–474.
- [9] Martin KM Engqvist et al. “Directed evolution of Gloeobacter violaceus rhodopsin spectral properties”. In: *Journal of molecular biology* 427.1 (2015), pp. 205–220.
- [10] Philip Gage. “A new algorithm for data compression”. In: *The C Users Journal* 12.2 (1994), pp. 23–38.
- [11] Raia Hadsell, Sumit Chopra, and Yann LeCun. “Dimensionality reduction by learning an invariant mapping”. In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06)*. Vol. 2. IEEE. 2006, pp. 1735–1742.
- [12] Sepp Hochreiter, Martin Heusel, and Klaus Obermayer. “Fast model-based protein homology detection without alignment”. In: *Bioinformatics* 23.14 (2007), pp. 1728–1736.
- [13] Elad Hoffer and Nir Ailon. “Deep metric learning using triplet network”. In: *International Workshop on Similarity-Based Pattern Recognition*. Springer. 2015, pp. 84–92.
- [14] Jeremy Howard and Sebastian Ruder. “Universal language model fine-tuning for text classification”. In: *arXiv preprint arXiv:1801.06146* (2018).
- [15] Junlin Hu, Jiwen Lu, and Yap-Peng Tan. “Deep metric learning for visual tracking”. In: *IEEE Transactions*

- on *Circuits and Systems for Video Technology* 26.11 (2015), pp. 2056–2068.
- [16] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. “Siamese neural networks for one-shot image recognition”. In: *ICML deep learning workshop*. Vol. 2. 2015.
  - [17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.
  - [18] Taku Kudo. “Subword regularization: Improving neural network translation models with multiple subword candidates”. In: *arXiv preprint arXiv:1804.10959* (2018).
  - [19] BG Kumar, Gustavo Carneiro, Ian Reid, et al. “Learning local image descriptors with deep siamese and triplet convolutional networks by minimising global loss functions”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 5385–5394.
  - [20] Himabindu Lakkaraju, Stephen H Bach, and Jure Leskovec. “Interpretable decision sets: A joint framework for description and prediction”. In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 2016, pp. 1675–1684.
  - [21] Quoc Le and Tomas Mikolov. “Distributed representations of sentences and documents”. In: *International conference on machine learning*. 2014, pp. 1188–1196.
  - [22] Mark Lennox, Neil Robertson, and Barry Devereux. “Expanding the Vocabulary of a Protein: Application of Subword Algorithms to Protein Sequence Modelling”. In: *The Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (2020).
  - [23] Benjamin Letham et al. “Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model”. In: *The Annals of Applied Statistics* 9.3 (2015), pp. 1350–1371.
  - [24] Zachary C Lipton. “The mythos of model interpretability”. In: *Queue* 16.3 (2018), pp. 31–57.
  - [25] Jiwen Lu et al. “Multi-manifold deep metric learning for image set classification”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1137–1145.
  - [26] Laurens van der Maaten and Geoffrey Hinton. “Visualizing data using t-SNE”. In: *Journal of machine learning research* 9.Nov (2008), pp. 2579–2605.
  - [27] Seonwoo Min et al. “Pre-Training of Deep Bidirectional Protein Sequence Representations with Structural Information”. In: *arXiv preprint arXiv:1912.05625* (2019).
  - [28] Tsendsuren Munkhdalai and Hong Yu. “Meta networks”. In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org. 2017, pp. 2554–2563.
  - [29] Matthew E Peters et al. “Deep contextualized word representations”. In: *arXiv preprint arXiv:1802.05365* (2018).
  - [30] Alec Radford et al. “Improving language understanding by generative pre-training”. In: *URL https://s3-us-west-2.amazonaws.com/openai-assets/researchcovers/languageunsupervised/language_understanding_paper.pdf* (2018).
  - [31] Roshan Rao et al. “Evaluating Protein Transfer Learning with TAPE”. In: *arXiv preprint arXiv:1906.08230* (2019).
  - [32] Carl Edward Rasmussen. “Gaussian processes in machine learning”. In: *Summer School on Machine Learning*. Springer. 2003, pp. 63–71.
  - [33] Philip A Romero, Andreas Krause, and Frances H Arnold. “Navigating the protein fitness landscape with Gaussian processes”. In: *Proceedings of the National Academy of Sciences* 110.3 (2013), E193–E201.
  - [34] Florian Schroff, Dmitry Kalenichenko, and James Philbin. “Facenet: A unified embedding for face recognition and clustering”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 815–823.
  - [35] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* (2014).
  - [36] Jake Snell, Kevin Swersky, and Richard Zemel. “Prototypical networks for few-shot learning”. In: *Advances in Neural Information Processing Systems*. 2017, pp. 4077–4087.
  - [37] Christian Szegedy et al. “Inception-v4, inception-resnet and the impact of residual connections on learning”. In: *Thirty-first AAAI conference on artificial intelligence*. 2017.
  - [38] Oriol Vinyals et al. “Matching networks for one shot learning”. In: *Advances in neural information processing systems*. 2016, pp. 3630–3638.
  - [39] Kilian Q Weinberger and Lawrence K Saul. “Distance metric learning for large margin nearest neighbor classification”. In: *Journal of Machine Learning Research* 10.Feb (2009), pp. 207–244.
  - [40] Zachary Wu et al. “Machine learning-assisted directed protein evolution with combinatorial libraries”. In: *Proceedings of the National Academy of Sciences* 116.18 (2019), pp. 8852–8858.
  - [41] Kevin K Yang, Zachary Wu, and Frances H Arnold. “Machine-learning-guided directed evolution for protein engineering”. In: *Nature methods* 16.8 (2019), pp. 687–694.
  - [42] Kevin K Yang et al. “Learned protein embeddings for machine learning”. In: *Bioinformatics* 34.15 (2018), pp. 2642–2648.
  - [43] Jason Yosinski et al. “How transferable are features in deep neural networks?” In: *Advances in neural information processing systems*. 2014, pp. 3320–3328.
  - [44] Julian Zaugg et al. “Learning epistatic interactions from sequence-activity data to predict enantioselectivity”. In: *Journal of computer-aided molecular design* 31.12 (2017), pp. 1085–1096.