Optimizers for Use with Deep Canonical Correlation Analysis

Geerthan Srikantharajah
Department of Electrical and Computer Engineering
Ryerson University

***Abstract*—The combination of neural networks and older techniques unlocks a lot of potential. Canonical correlation analysis, an older technique that correlates data samples, can be significantly improved. There are many variants of canonical correlation analysis, and one variant involves the use of two neural networks: deep canonical correlation analysis. This paper focuses on using deep canonical correlation analysis to identify phonetics based on acoustic and articulatory data. This paper compares the use of two different optimizers, stochastic gradient descent (SGD) and L-BFGS, as two published papers have varying results. Experiments show that SGD is better than L-BFGS when configured correctly.**

I. Introduction

Data analysis is a very hot topic, with the rise of machine learning and the availability of large datasets. However, data analysis was still possible years ago. The major difference is that machine learning allows for more complex problems to be solved, and while neural networks were researched a long time ago, we lacked the computational power to utilize them until recently. One example of an ML-dominated field is computer vision. Handcrafted feature descriptors and basic image manipulation (ex. thresholding) can be powerful when used effectively, but most of the latest research focuses on neural networks. You only look once (YOLO) [1] is an example of a very powerful object detection algorithm that uses a neural network. Complicated image manipulation and fine-tuned values for every step are not required with this type of solution. A few values still need to be fine-tuned, and these pertain to the network: the batch size and learning rate are examples of hyperparameters. Additionally, different network structures can lead to varying results. A rule of thumb is that neural networks are not one-size-fits-all.

One way to improve the effectiveness of a neural network is to use it in conjunction with other techniques. Neural networks do not have to be the entire solution; they can be a component instead. A lot of older research has been buried by the rise of ML, so understanding this research is important to further the effectiveness of ML. This paper will focus on the fusion of neural networks with an older process called canonical correlation analysis (CCA) [2].

Canonical correlation analysis is a statistical process used on two sets of data to obtain highly correlated samples. It is possible to perform matching with CCA, as matching data can have a high correlation if implemented correctly. An example of a matching problem would be face matching, where each sample could represent a 2D image of a face, or a vector of extracted features from that image. Another way to use CCA is to combine it with a secondary matching algorithm. The detection of faces is still possible in this way. It is common to have two sets of data, where each sample will have one representation in each set of data. Each sample contains the same data, but with separate representations. After performing canonical correlation analysis on a training set of data, new samples can be matched to 'learned' faces based on their distance to the CCA-transformed training data.

Basic canonical correlation analysis is referred to as linear canonical correlation analysis or CCA. Linear CCA is very limited, both in terms of correlation detection (linear) and flexibility (two-set restriction). Many variations of CCA have been researched over the years, to improve on these areas. The variant responsible for lifting the two-set restriction is multiple canonical correlation analysis (MCCA) [3]. Multiple CCA adds a world of additional possibilities to canonical correlation analysis, and is helpful for multimodal/multiview data. An example of multiview data would be a triple-camera setup pointed at the same position. Multimodal data brings CCA closer to human perception, as several modes can be added, similar to our five senses. MCCA is key for various other CCA methods, as they use the principles that MCCA employs in order to extend their own methods into multi-set ones. However, two-set analysis still goes a long way. Deep canonical correlation analysis (DCCA) [4], [5] is the technique that will be focused on within this paper. Deep CCA alleviates the correlation detection weakness that CCA has, with support for non-linear correlation detection through two neural networks. There is a multiset extension of deep CCA (Deep Multiset Canonical Correlation Analysis, [6]), but this paper will focus on the two-set problem.

Multimodal data is well suited for canonical correlation analysis. Each modality contains a large amount of information on its own, and the combination of modalities brings the analysis closer to a real-life perception of the data. This paper will focus on the analysis of the University of Wisconsin X-Ray Microbeam Speech Production Database (XRMB) [5]. The XRMB dataset contains acoustic and articulatory data of human speech. For each sample, the speaker and spoken phonetic is provided. The DCCA works by Andrew *et al.* [4] and Wang *et al.* [5] both attempt to identify the spoken phonetic based off of the acoustic and articulatory data of different speakers. They set up their neural networks in different ways, and achieved different results. The work by Andrew *et al.* had no luck with stochastic gradient descent (SGD) and resorted to

the Limited-Memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) optimizer, a full batch optimizer. The work by Wang *et al.* obtained better results with stochastic gradient descent instead, using mini-batches. This paper will compare the two approaches on the XRMB dataset, removing as many factors as possible.

II. Related Work

A. Linear Canonical Correlation Analysis

Canonical correlation analysis is a very old topic, as Hotelling wrote about it back in 1936. However, it is a powerful tool when used properly. Given two sets of data, with paired samples, CCA aims to create linear projections of each dataset that are highly correlated with each other. The linear projections are reduced in dimension from the original set with dimensions of o and p, to a dimension of d. Given the input sets $x_1 \in \mathbb{R}^{o \times n}$, $x_2 \in \mathbb{R}^{p \times n}$ where n indicates n samples, if $d = 1$, the projected sets would be equal to $X_i \in \mathbb{R}^{1 \times n}$, $X_i = \omega_i^T * x_i$. In this equation, ω_i is a set of linear coefficients (canonical weight vectors) combined with set i in order to generate one canonical variate in X_i. It is normal for d pairs of ω_1, ω_2 canonical weight vectors to be used in order to generate canonical variates of the format $X_i \in \mathbb{R}^{d \times n}$. The amount of canonical variates cannot exceed the original dimensionality of either input set. The CCA problem can be solved by representing it as the generic eigenvalue (GEV) problem, shown in equation (1). In the equation, R_{x_1,x_2} indicates the cross-correlation matrix between x_1 and x_2, and μ represents the canonical correlation score.

$$\begin{bmatrix} 0 & R_{x_1,x_2} \\ R_{x_2,x_1} & 0 \end{bmatrix} \omega = \mu \begin{bmatrix} R_{x_1,x_1} & 0 \\ 0 & R_{x_2,x_2} \end{bmatrix} \omega \quad (1)$$

B. Canonical Correlation Analysis Variants

Many of the canonical correlation analysis variants are two-set variants. Generally, the two-set variants of CCA aim to put more important data into the original CCA function shown in equation (1) so that it can all be solved as the GEV problem. Discriminative canonical correlation analysis (dCCA) [7] is an example of a two-set CCA variant. The additional data that dCCA aims to use is discriminative data. If samples from different classes had a shared similarity, CCA would identify this as a correlation between different classes. Discriminative CCA uses class data in its computation, maximizing within-class correlations while minimizing between-class correlations. This change makes a big difference when performing class matching with CCA. The GEV problem of dCCA is shown in equation (2), where S_b is the between-class correlation, S_w is the within-class correlation, and T is the resulting transformation matrix.

$$S_b T = \lambda S_w T \quad (2)$$

An example of an unconventional CCA method is two dimensional canonical correlation analysis (2DCCA) [8]. Image features are commonly used instead of raw images during processing, as they extract the most important parts. Processing an entire image is very difficult to do properly. One major problem is that images get flattened to one dimension when used in CCA, meaning that spatial data is lost. An image of size (x, y) would be the same as an image of size (y, x), if the pixels were in the same order. 2DCCA reformulates the CCA problem such that inputs can be provided as two 2D matrices. The input data is directly used, meaning that no spatial data is lost.

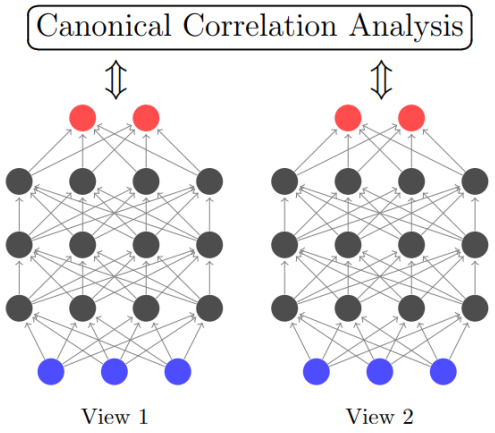

Fig. 1: The DCCA layout

Multiple canonical correlation analysis (MCCA) [3] attempts to fix the two-set limitation that CCA has. There are several variants of MCCA, but the most common one aims to minimize the sum of correlations between each projection. The correlations between every pair of sets' projected data is used for this calculation. MCCA is commonly combined with other techniques to extend them to the multi-set case. Discriminative multiple canonical correlation analysis (DMCCA) [9] is an example of this extension. The dCCA formula in equation (2) is recreated with larger matrices containing data from multiple sets. It is still mapped as the GEV problem, and can be solved in the same way.

2D canonical correlation analysis can also be adapted to the multi-set case. While it is a unique CCA technique, it still maps the equations as GEV problems (as 2DCCA uses two GEV problems). The matrices used as input to these problems can simply be expanded to contain data for several input sets.

The aforementioned variants aimed to solve different weaknesses of linear canonical correlation analysis, but they are all linear techniques. Kernel canonical correlation analysis (KCCA) is a nonlinear canonical correlation analysis technique. By using the kernel trick (using Hilbert space representations of the input sets), it is possible to create two GEV problems, one per set.

III. Deep Canonical Correlation Analysis

Deep canonical correlation analysis (DCCA) [4], [5] is a two-set non-linear CCA technique (similar to KCCA). However, DCCA uses a very different structure compared to all of the other CCA techniques mentioned above. In Figure (1), you can see that DCCA is based on two deep neural networks

(DNNs), one per input set. The DNN structures are configurable, as the number of hidden layers as well as the number of nodes per layer are hyperparameters. Each DNN is responsible for projecting input samples into canonical variates, meaning that there are no canonical weight vectors. The GEV problem is not used within the transformation process. Let n represent the number of samples, o represent the dimensionality of input set 1, and p represent the dimensionality of input set 2. Let the input sets be represented as $x_1 \in \mathbb{R}^{o \times n}$ and $x_2 \in \mathbb{R}^{p \times n}$, θ_i contain all parameters of the ith neural network, and θ_i^* represent the ideal θ_i. The equation for θ_i^* is shown below.

$$(\theta_1^*, \theta_2^*) = \underset{(\theta_1, \theta_2)}{\operatorname{argmax}} \operatorname{corr}(f_1(x_1; \theta_1), f_2(x_2; \theta_2)) \tag{3}$$

The correlation requires additional calculations. Given the two resulting matrices $H_1, H_2 \in \mathbb{R}^{o \times m}$ from each DNN after centering, equations (4-7) are used. In equation (4), r_i is a small regularization constant, and I is an identity matrix.

$$\hat{\Sigma}_{ii} = \frac{1}{m-1} H_i H_i^{'} + r_i I \tag{4}$$

$$\hat{\Sigma}_{12} = \frac{1}{m-1} H_1 H_2^{'} \tag{5}$$

$$T = \hat{\Sigma}_{11}^{-1/2} \hat{\Sigma}_{12} \hat{\Sigma}_{22}^{-1/2} \tag{6}$$

$$\operatorname{corr}(H_1, H_2) = \operatorname{tr}(T^{'} T)^{1/2} \tag{7}$$

Andrew *et al.* used a denoising autoencoder for parameter initialization, as a form of pre-training.

The study by Wang *et al.* found contradictory results to Andrew *et al.*, with several differing components in the implementations. Andrew *et al.* could not find satisfactory results with a mini-batch based optimizer (stochastic gradient descent (SGD)), and resorted to a full batch optimizer (L-BFGS). Wang *et al.* claimed that SGD worked better than L-BFGS, with a large enough mini-batch size. The inconsistencies do not end here.

The use of a non-linear activation function is crucial, but the two studies used different activation functions. The study by Andrew *et al.* created a custom sigmoid function, which does not plateau at high/low values. It is termed a non-saturating sigmoid function. The study by Wang *et al.* used ReLU activation instead, citing that it was faster, with a similar level of performance.

Within the test that Wang *et al.* performed, many components were inconsistent. The network was initialized randomly (instead of using a denoising autoencoder), layer depth and hidden node counts were changed, and the network used a few hyperparameters that were automatically configured through a grid search (r_i, learning rate, and momentum). Most of these changes are beneficial for accuracy, but consistency between tests is lost. The outlier is random parameter initialization, which removes the bias that pre-training could have.

IV. Experiment

The purpose of this paper is to evaluate the two differing optimizers, SGD and L-BFGS, when used on deep CCA. The dataset that this test is conducted on is the University of Wisconsin X-Ray Microbeam Speech Production Database (XRMB) [5]. Both DCCA papers also performed tests on the XRMB dataset. The XRMB dataset contains approximately 1.4 million samples measuring human speech in two sets, an acoustic set and an articulatory set. The acoustic set has a dimensionality of 273, and the articulatory set has a dimensionality of 112. Each sample also contains two labels, one for the speaker, and one for the phonetic that was pronounced. There are 47 speakers and 38 phonetics in the dataset. The XRMB dataset is a good multimodal dataset which benefits from nonlinear CCA, as shown in Andrew *et al.*'s results.

This particular experiment will compare the SGD and L-BFGS optimizers for use with deep CCA, keeping all other factors constant. The network was not initialized with a denoising autoencoder and hyperparameters were all manually set. Both DNNs had 2 hidden layers, but the acoustic set's DNN had 1800 hidden nodes per layer while the articulatory set's DNN had 1200 hidden nodes per layer. ReLU activation was used as well. The network structure was set up to match Wang *et al.*'s tests.

Along with the network parameter initialization and hyperparameter initialization changes, there was another modification to the test. Wang *et al.* compared the total correlation values, and used them as a benchmark. While total correlation values are a good metric, it is better to use classification accuracy. Better total correlation does not always indicate better classification accuracy.

Only the samples from speaker 11 are considered for this test. This is in line with Wang *et al.*'s tests. L-BFGS is a very memory-intensive optimizer, so it is difficult to add more data to the test. There is no configurable batch size either, as it is a full batch optimizer.

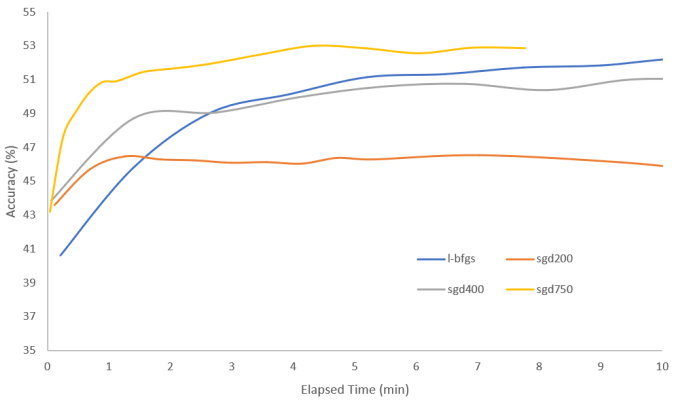

Fig. 2: Graph of Accuracy over Time

Tests were run on a Google Colab instance with an NVIDIA Tesla K80 GPU. They were compared by runtime. The figure above shows the results. Each SGD line represents SGD running at a set batch size. For example, sgd200 means SGD with a batch size of 200. SGD200 is not good past the 2 minute mark relative to other options, but it still outperforms

L-BFGS at the very beginning. SGD-750 is very good overall. This matches Wang *et al.*'s results. We can see that L-BFGS certainly takes longer to build accuracy, but it performs well past the 3 minute mark. In the original study, L-BFGS underperformed in the long term when compared to most of their SGD tests. With most of the variances removed, we see that L-BFGS performs fine in the long run when compared to SGD. However, SGD with an optimal batch size will still outperform L-BFGS.

V. Conclusion

This study examined deep canonical correlation analysis and its performance on the X-Ray Microbeam Dataset (XRMB) using two different optimizers, stochastic gradient descent and L-BFGS. It was observed that when most variances were removed from deep CCA, SGD would only outperform L-BFGS with certain batch sizes in the long run, unlike what was seen in [5]. However, when the model is first trained, SGD is significantly better than L-BFGS at all tested batch sizes. L-BFGS also uses significantly more memory. The tests had to be altered purely because of its memory footprint. When using DCCA on the XRMB dataset, SGD with the correct batch size is a clear favourite, but selecting the correct batch size is important. Future research includes training on the XRMB dataset with more speakers (only one was used due to the memory limitations), and evaluating DCCA on other datasets.

References

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2016, pp. 779–788. doi: 10.1109/CVPR.2016.91.

[2] H. Hotelling, "RELATIONS BETWEEN TWO SETS OF VARIATES," Biometrika, vol. 28, no. 3–4, pp. 321–377, Dec. 1936, doi: 10.1093/biomet/28.3-4.321.

[3] J. R. Kettenring, "Canonical analysis of several sets of variables," Biometrika, vol. 58, no. 3, pp. 433–451, 1971, doi: 10.1093/biomet/58.3.433.

[4] G. Andrew, R. Arora, J. Bilmes, and K. Livescu, "Deep Canonical Correlation Analysis," in Proceedings of the 30th International Conference on Machine Learning, May 2013, pp. 1247–1255. Accessed: Apr. 30, 2022. [Online]. Available: https://proceedings.mlr.press/v28/andrew13.html

[5] W. Wang, R. Arora, K. Livescu, and J. A. Bilmes, "Unsupervised learning of acoustic features via deep canonical correlation analysis," in 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), South Brisbane, Queensland, Australia, Apr. 2015, pp. 4590–4594. doi: 10.1109/ICASSP.2015.7178840.

[6] K. Somandepalli, N. Kumar, R. Travadi, and S. Narayanan, "Multimodal Representation Learning using Deep Multiset Canonical Correlation," arXiv:1904.01775 [cs, eess], Apr. 2019, Accessed: Apr. 30, 2022. [Online]. Available: http://arxiv.org/abs/1904.01775

[7] T.-K. Kim, J. Kittler, and R. Cipolla, "Discriminative Learning and Recognition of Image Set Classes Using Canonical Correlations," IEEE Trans. Pattern Anal. Mach. Intell., vol. 29, no. 6, pp. 1005–1018, Jun. 2007, doi: 10.1109/TPAMI.2007.1037.

[8] S. H. Lee and S. Choi, "Two-Dimensional Canonical Correlation Analysis," IEEE Signal Processing Letters, vol. 14, no. 10, pp. 735–738, Oct. 2007, doi: 10.1109/LSP.2007.896438.

[9] L. Gao, L. Qi, E. Chen, and L. Guan, "Discriminative Multiple Canonical Correlation Analysis for Multi-feature Information Fusion," in 2012 IEEE International Symposium on Multimedia, Dec. 2012, pp. 36–43. doi: 10.1109/ISM.2012.15.