


Causal Deep Learning - Supplemental

M. Alex O. Vasilescu 

IPAM, University of California, Los Angeles CA, USA
Tensor Vision, Los Angeles CA, USA

Notation

We denote scalars by lower case italic letters (a, b, \dots), vectors by bold lower case letters \mathbf{a}, \mathbf{b} etc., matrices by bold uppercase letters ($\mathbf{A}, \mathbf{B}, \dots$), and higher-order tensors by bold uppercase calligraphic letters ($\mathcal{A}, \mathcal{B}, \dots$). Index upper bounds are denoted by italic uppercase letters (i.e., $1 \leq a \leq A$ or $1 \leq i \leq I$). The zero matrix is denoted by $\mathbf{0}$, and the identity matrix is denoted by \mathbf{I} .

a, b, \dots	scalars - lower case italic
$1 \leq a \leq A, 1 \leq i \leq I, \dots$	scalar upper bounds - upper case italic
$\mathbf{a}, \mathbf{b}, \dots$	vectors - lower case bold
$\mathbf{A}, \mathbf{B}, \dots$	matrices - upper case bold
$\mathbf{0}, \mathbf{I}$	zero matrix and identity matrix
$\mathcal{A}, \mathcal{B}, \dots$	higher-order tensors - calligraphic

A. PCA computation with a Hebb autoencoder

A Hebb autoencoder-decoder [8] with a linear transfer function, Fig. 1, minimizes the least squares function,

$$L = \sum_{i=1}^I \|\mathbf{d}_i - \mathbf{B}\mathbf{c}_i\|^2 + \lambda \|\mathbf{B}^T \mathbf{B} - \mathbf{I}\|. \quad (1)$$

where $\{\mathbf{d}_i \in \mathbb{C}^{I_x} | 1 \leq i \leq I\}$ is a set of I different vectorized and mean centered observations with I_x measurements, $\mathbf{B} \in \mathbb{C}^{I_x \times R}$ is the PCA basis matrix, and \mathbf{c}_i is the representation of \mathbf{d}_i relative to \mathbf{B} [5, p. 58]. The columns \mathbf{b}_r in \mathbf{B} are sequentially computed, and their contributions are subtracted from a vectorized centered training data set. The remainder is modeled by the next basis vector \mathbf{b}_{r+1} , i.e., the next Hebb neuron. The weights of the r^{th} Hebb neuron are the elements in \mathbf{b}_r which are updated using natural gradient descent [5, p. 58]. The update rule is also known as the Sanger Rule [19,18,17,1,15] in machine learning,

* "Causal Deep Learning" to appear ICPR 2024. LNCS, vol 15309, pg.420-438. Springer, Cham. https://doi.org/10.1007/978-3-031-78189-6_27. First presented at ICPR'22, Aug'22, Montreal

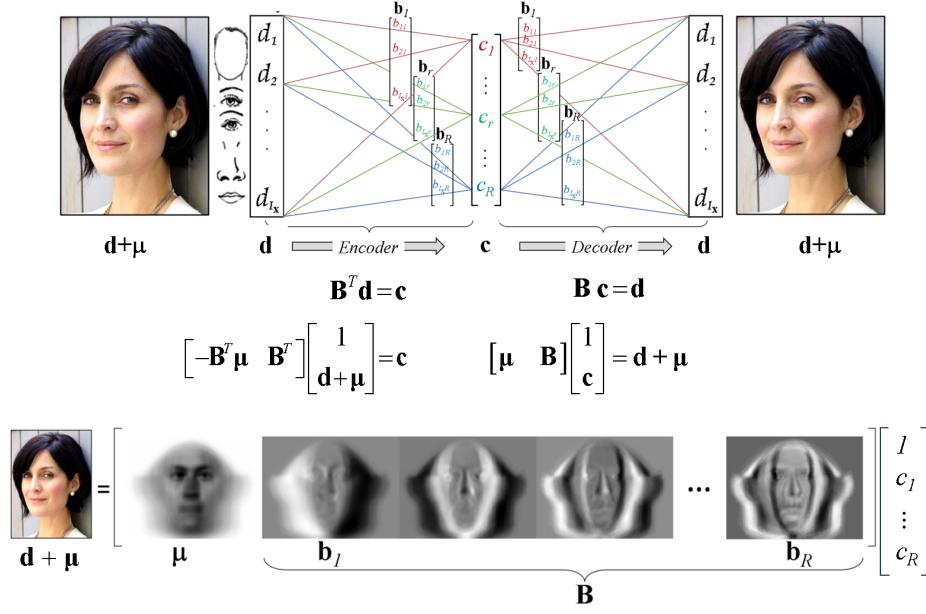


Fig. 1: Autoencoder-decoder architecture and Principal Component Analysis. The basis vector \mathbf{b}_r is a columns in \mathbf{B} and its contribution c_r is an element in \mathbf{c} . (All images have been vectorized, but they are displayed as a grid of numbers.)

$$\Delta \mathbf{b}_r(t+1) = \eta \left(\mathbf{d} - \sum_{i_r=1}^r \mathbf{b}_{i_r}(t) c_{i_r}(t) \right) c_r(t) \quad (2)$$

$$= \eta \left(\mathbf{d} - \sum_{i_r=1}^r \mathbf{b}_{i_r}(t) \mathbf{b}_{i_r}^T(t) \mathbf{d} \right) (\mathbf{b}_r(t)^T \mathbf{d})^T \quad (3)$$

$$\mathbf{b}_r(t+1) = \frac{\mathbf{b}_r(t) + \Delta \mathbf{b}_r(t+1)}{\|\mathbf{b}_r(t) + \Delta \mathbf{b}_r(t+1)\|}, \quad (4)$$

where $0 \leq \eta \leq 2/\|\mathbf{B}\|^2 = \sigma_{\max, \mathbf{B}}$ is the learning rate, c_r is the contribution of \mathbf{b}_r , and t is the time iteration. Back-propagation[12,13] performs PCA gradient descent. An autoencoder may be trained and the weights updated using a set of data batches, $\{\mathbf{D}_i\}$,

$$\Delta \mathbf{b}_r(t+1) = \eta \left(\mathbf{D}_i - \sum_{i_r=1}^r \mathbf{b}_{i_r}(t) \mathbf{b}_{i_r}^T(t) \mathbf{D}_i \right) (\mathbf{b}_r(t)^T \mathbf{D}_i)^T \quad (5)$$

$$= \eta \left(\mathbf{D}_i \mathbf{D}_i^T - \sum_{i_r=1}^r \mathbf{b}_{i_r}(t) \mathbf{b}_{i_r}^T(t) \mathbf{D}_i \mathbf{D}_i^T \right) \mathbf{b}_r(t) \quad (6)$$

$$= \eta (\mathbf{D}_i \mathbf{D}_i^T - \mathbf{B}_r(t) \mathbf{B}_r^T(t) \mathbf{D}_i \mathbf{D}_i^T) \mathbf{b}_r(t), \quad (7)$$

where \mathbf{B}_r contains the first r columns. Computational speed-ups and better solutions are achieved with stochastic gradient descent [3][16].

B. Relevant Tensor Algebra

Briefly, tensors are the natural generalization of matrices–linear operators defined over a vector space that map a data point from one vector space to another with preferred properties. Tensors define multilinear operators over a *set* of vector spaces.

Definition 1 (Tensor). *Tensors are multilinear mappings over a set of domain vector spaces, \mathbb{C}^{I_m} , $1 \leq m \leq M$, to a range vector space \mathbb{C}^{I_0} :*

$$\mathcal{A} : \{\mathbb{C}^{I_1} \times \mathbb{C}^{I_2} \times \dots \times \mathbb{C}^{I_M}\} \mapsto \mathbb{C}^{I_0}. \quad (8)$$

The order of tensor $\mathcal{A} \in \mathbb{C}^{I_0 \times I_1 \times \dots \times I_M}$ is $M + 1$. An element of \mathcal{A} is denoted as $\mathcal{A}_{i_0 i_1 \dots i_M}$ or $a_{i_0 i_1 \dots i_M}$, where $1 \leq i_m \leq I_m$.

In a causal tensor framework, the M domain spaces span the causal factor representations and the range vector space spans the observation space. An M -way data array is informally referred to as a “*data tensor*”.

The mode- m vectors of an M -order tensor $\mathcal{A} \in \mathbb{C}^{I_0 \times I_1 \times \dots \times I_M}$ are the I_m -dimensional vectors obtained from \mathcal{A} by varying index i_m while keeping the other indices fixed. In tensor terminology, column vectors are the mode-0 vectors and row vectors as mode-1 vectors. The mode- m vectors of a tensor are also known as *fibers*. The mode- m vectors are the column vectors of matrix $\mathbf{A}_{[m]}$ that results from *matrixizing* (a.k.a. *flattening*) the tensor \mathcal{A} .

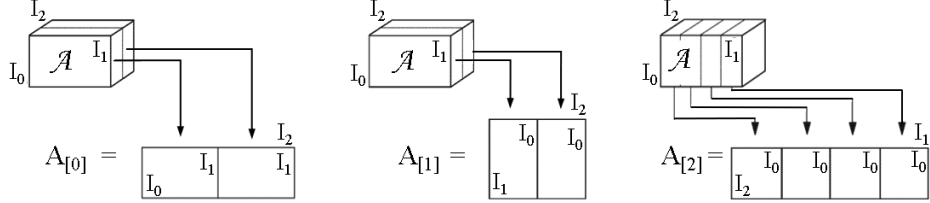


Fig. 2: Matrixizing a 3rd order tensor, \mathcal{A} .

Definition 2 (Mode- m Matrixizing). *The mode- m matrixizing of tensor $\mathcal{A} \in \mathbb{C}^{I_0 \times I_1 \times \dots \times I_M}$ is defined as the matrix $\mathbf{A}_{[m]} \in \mathbb{C}^{I_m \times (I_0 \dots I_{m-1} I_{m+1} \dots I_M)}$. As the parenthetical ordering indicates, the mode- m column vectors are arranged by sweeping all the other mode indices through their ranges, with smaller mode indexes varying more rapidly than larger ones; thus,*

$$[\mathbf{A}_{[m]}]_{jk} = a_{i_1 \dots i_m}, \quad \text{where} \quad (9)$$

$$j = i_m \quad \text{and} \quad k = 1 + \sum_{\substack{n=0 \\ n \neq m}}^M (i_n - 1) \prod_{\substack{l=0 \\ l \neq m}}^{n-1} I_l.$$

Algorithm 1 M -mode SVD algorithm.[21]**Input** the data tensor $\mathcal{D} \in \mathbb{C}^{I_0 \times \dots \times I_M}$.

1. For $m := 0, \dots, M$,
Let \mathbf{U}_m be the left orthonormal matrix of $[\mathbf{U}_m \mathbf{S}_m \mathbf{V}_m^T] := \text{svd}(\mathbf{D}_{[m]})^a$
2. Set $\mathcal{Z} := \mathcal{D} \times_0 \mathbf{U}_0^T \times_1 \mathbf{U}_1^T \cdots \times_m \mathbf{U}_m^T \dots \times_M \mathbf{U}_M^T$.

Output mode matrices $\mathbf{U}_0, \mathbf{U}_1, \dots, \mathbf{U}_M$, and the core tensor \mathcal{Z} .

^a The computation of \mathbf{U}_m in the SVD $\mathbf{D}_{[m]} = \mathbf{U}_m \mathbf{\Sigma} \mathbf{V}_m^T$ can be performed efficiently, depending on which dimension of $\mathbf{D}_{[m]}$ is smaller, by decomposing either $\mathbf{D}_{[m]} \mathbf{D}_{[m]}^T = \mathbf{U}_m \mathbf{\Sigma}^2 \mathbf{U}_m^T$ (note that $\mathbf{V}_m^T = \mathbf{\Sigma}^+ \mathbf{U}_m^T \mathbf{D}_{[m]}$) or by decomposing $\mathbf{D}_{[m]}^T \mathbf{D}_{[m]} = \mathbf{V}_m \mathbf{\Sigma}^2 \mathbf{V}_m^T$ and then computing $\mathbf{U}_m = \mathbf{D}_{[m]} \mathbf{V}_m \mathbf{\Sigma}^+$.

A generalization of the product of two matrices is the product of a tensor and a matrix [6,4].

Definition 3 (Mode- m Product, \times_m). *The mode- m product of a tensor $\mathcal{A} \in \mathbb{C}^{I_1 \times \dots \times I_m \times \dots \times I_M}$ and a matrix $\mathbf{B} \in \mathbb{C}^{J_m \times I_m}$, denoted by $\mathcal{A} \times_m \mathbf{B}$, and it results in a tensor of dimensionality $\mathbb{C}^{I_1 \times \dots \times I_{m-1} \times J_m \times I_{m+1} \times \dots \times I_M}$ whose entries are computed by*

$$[\mathcal{A} \times_m \mathbf{B}]_{i_1 \dots i_{m-1} j_m i_{m+1} \dots i_M} = \sum_{i_m} a_{i_1 \dots i_{m-1} i_m i_{m+1} \dots i_M} b_{j_m i_m},$$

$$\mathcal{C} = \mathcal{A} \times_m \mathbf{B} \xrightleftharpoons[\text{tensorize}]{\text{matrixize}} \mathbf{C}_{[m]} = \mathbf{B} \mathbf{A}_{[m]}.$$

The M -mode SVD, Algorithm 1 proposed by Vasilescu and Terzopoulos [26] is a “generalization” of the conventional matrix (i.e., 2-mode) SVD which may be written in tensor notation as

$$\mathbf{D} = \mathbf{U}_0 \mathbf{S} \mathbf{U}_1^T \quad \Leftrightarrow \quad \mathbf{D} = \mathbf{S} \times_0 \mathbf{U}_0 \times_1 \mathbf{U}_1$$

The M -mode SVD orthogonalizes the M spaces and decomposes a tensor as the *mode- m product*, denoted \times_m , of M -orthonormal mode matrices, and a core tensor \mathcal{Z}

$$\mathcal{D} = \mathcal{Z} \times_0 \mathbf{U}_0 \cdots \times_m \mathbf{U}_m \cdots \times_M \mathbf{U}_M. \quad (10)$$

$$\mathbf{D}_{[m]} = \mathbf{U}_m \mathbf{Z}_{[m]} (\mathbf{U}_M \cdots \otimes \mathbf{U}_{m+1} \otimes_{m-1} \mathbf{U} \cdots \otimes \mathbf{U}_0)^T, \quad (11)$$

$$\text{vec}(\mathcal{D}) = (\mathbf{U}_M \cdots \otimes \mathbf{U}_{m+1} \otimes \mathbf{U}_{m-1} \cdots \otimes \mathbf{U}_0) \text{vec}(\mathcal{Z}). \quad (12)$$

The latter two equations express the decomposition in matrix form and in terms of *vec* operators.

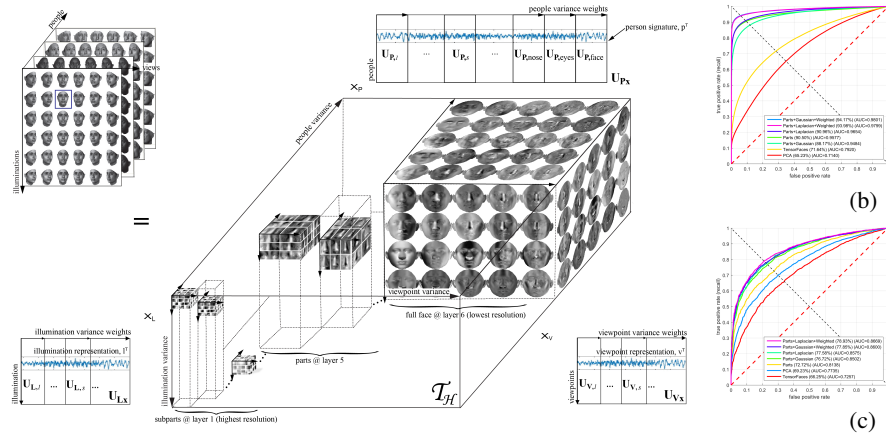


Fig. 3: Compositional Hierarchical Block TensorFaces [24] learns a hierarchy of features, and represents each person as a part-based compositional representation. Figure depicts the training data factorization, $\mathcal{D} = \mathcal{T}_{\mathcal{H}} \times_L \mathbf{U}_L \times_V \mathbf{U}_V \times_P \mathbf{U}_P$, where an observation is represented as $\mathbf{d}(\mathbf{p}, \mathbf{v}, \mathbf{l}) = \mathcal{T}_{\mathcal{H}} \times_L \mathbf{l}^T \times_V \mathbf{v}^T \times_P \mathbf{p}^T$ and $\mathcal{T}_{\mathcal{H}}$ spans the hierarchical causal factor variance. (b) ROC curves for the University of Freiburg 3D Morphable Faces dataset. (c) ROC curves for the LFW dataset. The average accuracies are listed next to each method, along with the area under the curve (AUC). Parts refers to using compositional hierarchical Block TensorFaces models to separately analyze facial parts. Gaussian, Laplacian refers to using compositional hierarchical Block TensorFaces on a Gaussian/Laplacian data pyramid.

C. Compositional Hierarchical Block TensorFaces

Training Data: In our experiments, we employed gray-level facial training images rendered from 3D scans of 100 subjects. The scans were recorded using a CyberwareTM 3030PS laser scanner and are part of the 3D morphable faces database created at the University of Freiburg [2]. Each subject was combinatorially imaged in Maya from 15 different viewpoints ($\theta = -60^\circ$ to $+60^\circ$ in 10° steps on the horizontal plane, $\phi = 0^\circ$) with 15 different illuminations ($\theta = -35^\circ$ to $+35^\circ$ in 5° increments on a plane inclined at $\phi = 45^\circ$).

Data Preprocessing: Facial images were warped to an average face template by a piecewise affine transformation given a set of facial landmarks obtained by employing Dlib software [11,10,20,14,7]. Illumination was normalized with an adaptive contrast histogram equalization algorithm, but rather than performing contrast correction on the entire image, subtiles of the image were contrast normalized, and tiling artifacts were eliminated through interpolation. Histogram clipping was employed to avoid over-saturated regions.

Experiments: We ran five experiments with five facial part-based hierarchies from which a person representation was computed, Fig. 3. Each image, $\mathbf{d} \in \mathbb{R}^{I_0 \times 1}$, was convolved with a Gaussian and a Laplacian filter bank $\{\mathbf{H}_s\}_{s=1 \dots S}$ that contained five filters, $S = 5$. The filtered images, $\mathbf{d} \times_0 \mathbf{H}_s$, resulted in five facial part hierarchies composed

Training Dataset	Test Dataset	PCA	TensorFaces	Compositional Hierarchical Block TensorFaces				
				Pixels	Gaussian Pyramid	Weighted Gaussian Pyramid	Laplacian Pyramid	Weighted Laplacian Pyramid
Freiburg	Freiburg	65.23%	71.64%	90.50%	88.17%	94.17%	90.96%	93.98%
Freiburg	LFW grey level images	69.23% ±1.51	66.25% ±1.60	72.72% ±2.14	76.72% ±1.65	77.85% ±1.83	77.58% ±1.45	78.93% ±1.77

Table 1: Empirical results reported for Freiburg and Labeled Faces in the Wild (LFW) using PCA, TensorFaces and Compositional Hierarchical Block TensorFaces representations. Pixels denotes independent facial part analysis Gaussian/Laplacian use a multi resolution pyramid to analyze facial features at different scales. Weighted denotes a weighted composite signature.

Freiburg Experiment:

Train on Freiburg: 6 views ($\pm 60^\circ, \pm 30^\circ, \pm 5^\circ$); 6 illums ($\pm 60^\circ, \pm 30^\circ, \pm 5^\circ$), 45 people

Test on Freiburg: 9 views ($\pm 50^\circ, \pm 40^\circ, \pm 20^\circ, \pm 10^\circ, 0^\circ$), 9 illums ($\pm 50^\circ, \pm 40^\circ, \pm 20^\circ, \pm 10^\circ, 0^\circ$), 45 different people

Labeled Faces in the Wild (LFW) Experiment:

Models were trained on approximately half of one percent ($0.5\% < 1\%$) of the 4.4M images used to train DeepFace.

Train on Freiburg:

15 views ($\pm 60^\circ, \pm 50^\circ, \pm 40^\circ, \pm 30^\circ, \pm 20^\circ, \pm 10^\circ, \pm 5^\circ, 0^\circ$), 15 illums ($\pm 60^\circ, \pm 50^\circ, \pm 40^\circ, \pm 30^\circ, \pm 20^\circ, \pm 10^\circ, \pm 5^\circ, 0^\circ$), 100 people

Test on LFW: We report the mean accuracy and standard deviation across standard literature partitions [9], following the Unrestricted, labeled outside data supervised protocol.

of (i) independent pixel parts (ii) parts segmented from different layers of a Gaussian pyramid that were equally or (iii) unequally weighed, (iv) parts were segmented from a Laplacian pyramid that were equally or (v) unequally weighed.

The composite person signature was computed for every test image by employing the multilinear projection algorithm [23,27], and signatures were compared with a nearest neighbor classifier.

To validate the effectiveness of our system on real-world images, we report results on “LFW” dataset (LFW) [9]. This dataset contains 13,233 facial images of 5,749 people. The photos are unconstrained (*i.e.*, “in the wild”), and include variation due to pose, illumination, expression, and occlusion. The dataset consists of 10 train/test splits of the data. We report the mean accuracy and standard deviation across all splits in Table 1. Fig. 3(b-c) depicts the experimental ROC curves. We follow the supervised “Unrestricted, labeled outside data” framework.

Results: While we cannot celebrate closing the gap on human performance, our results are promising. DeepFace, a CNN model, improved the prior art verification rates on LFW from 70% to 97.35%, by training on 4.4M images of 200×200 pixels from 4,030 people, the same order of magnitude as the number of people in the LFW database.

We trained on less than one percent (1%) of the 4.4M total images used to train DeepFace. Images were rendered from 3D scans of 100 subjects with an the intraocular distance of approximately 20 pixels and with a facial region captured by 10,414 pixels (image size $\approx 100 \times 100$ pixels). We have currently achieved verification rates just shy of 80% on LFW.

Summary: Compositional Hierarchical Block TensorFaces models cause-and-effect as a hierarchical block tensor interaction between intrinsic and extrinsic causal factors of data formation [25][22].

A data tensor expressed as a part-based a hierarchy is a unified tensor model of wholes and parts. The resulting causal factor representations are interpretable, hierarchical, and statistically invariant to all other causal factors. While we have not closed the gap on human performance, we report encouraging face verification results on two test data sets—the Freiburg, and the Labeled Faces in the Wild datasets by training on a very small set of synthetic images. We have currently achieved verification rates just shy of eighty percent on LFW by employing synthetic images from 100 people, 15 viewpoints and 15 illuminations, for a total that constitutes less than one percent (1%) of the total images employed by DeepFace. CNN verification rates improved by 70% prior art to 97.35% only when they employed 4.4M images from 4,030 people, the same order of magnitude as the number of people in the LFW database.

References

1. D. H. Ackley, G. A. Hinton, and T. J. Sejnowski. A learning algorithm for Boltzmann machines. *Cognitive Science*, 9(1):147–169, 1985.
2. V. Blanz and T. A. Vetter. Morphable model for the synthesis of 3D faces. In *Proc. ACM SIGGRAPH 99 Conf.*, pages 187–194, 1999.
3. L. Bottou et al. Online learning and stochastic approximations. *On-line learning in neural networks*, 17(9):142, 1998.
4. J. D. Carroll, S. Pruzansky, and J. B. Kruskal. CANDELINC: A general approach to multidimensional analysis of many-way arrays with linear constraints on parameters. *Psychometrika*, 45:3–24, 1980.
5. C. Chatfield and A. Collins. *Introduction to Multivariate Analysis*, 1983.
6. L. de Lathauwer. *Signal Processing Based on Multilinear Algebra*. PhD dissertation, Katholieke Univ. Leuven, Belgium, 1997.
7. A. Hatamizadeh, D. Terzopoulos, and A. Myronenko. End-to-end boundary aware networks for medical image segmentation. In *Inter. Workshop on Machine Learning in Medical Imaging*, pages 187–194. Springer, 2019.
8. D. O. Hebb. *The organization of behavior: A neuropsychological theory*. John Wiley And Sons, Inc., New York, 1949.
9. G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, Oct 2007.
10. V. Kazemi and J. Sullivan. One millisecond face alignment with an ensemble of regression trees. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition, CVPR '14*, pages 1867–74, Washington, DC, USA, 2014. IEEE Computer Society.

11. D. E. King. Dlib-ml: A machine learning toolkit. Journal of Machine Learning Research, 10:1755–1758, 2009.
12. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, Nov 1998.
13. Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller. Efficient BackProp, pages 9–48. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
14. I. Macedo, E. V. Brazil, and L. Velho. Expression transfer between photographs through multilinear aam’s. pages 239–246, Oct 2006.
15. E. Oja. A simplified neuron model as a principal component analyzer. Journal of Mathematical Biology, 15:267–2735, 1982.
16. H. Robbins and S. Monro. A stochastic approximation method. The annals of mathematical statistics, pages 400–407, 1951.
17. D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. Parallel Distributed Processing, 1986.
18. T. Sanger. Optimal unsupervised learning in a single layer linear feedforward neural network. Neural Networks, 12:459–473, 1989.
19. T. Sejnowski, S. Chattarji, and P. Sfanton. Induction of Synaptic Plasticity by Hebbian Covariance in the Hippocampus, pages 105–124. Addison-Wesley, 1989.
20. W. Si, K. Yamaguchi, and M. A. O. Vasilescu. Face Tracking with Multilinear (Tensor) Active Appearance Models. Jun 2013.
21. M. A. O. Vasilescu. Human motion signatures: Analysis, Synthesis, Recognition. In Proc. Int. Conf. on Pattern Recognition, volume 3, pages 456–460, Quebec City, Aug 2002.
22. M. A. O. Vasilescu. Incremental Multilinear SVD. In Proc. Conf. on ThRee-way methods In Chemistry And Psychology (TRICAP 06), 2006.
23. M. A. O. Vasilescu. Multilinear projection for face recognition via canonical decomposition. In Proc. IEEE Inter. Conf. on Automatic Face Gesture Recognition (FG 2011), pages 476–483, Mar 2011.
24. M. A. O. Vasilescu and E. Kim. Compositional hierarchical tensor factorization: Representing hierarchical intrinsic and extrinsic causal factors. In The 25th ACM SIGKDD Conf. on Knowledge Discovery and Data Mining (KDD 2019): Tensor Methods for Emerging Data Science Challenges Workshop, Aug. 5 2019.
25. M. A. O. Vasilescu, E. Kim, and X. S. Zeng. CausalX: Causal eXplanations and block multilinear factor analysis. In 2020 25th International Conference of Pattern Recognition (ICPR 2020), pages 10736–10743, Jan 2021.
26. M. A. O. Vasilescu and D. Terzopoulos. Multilinear analysis of image ensembles: TensorFaces. In Proc. European Conf. on Computer Vision (ECCV 2002), pages 447–460, May 2002.
27. M. A. O. Vasilescu and D. Terzopoulos. Multilinear projection for appearance-based recognition in the tensor framework. In Proc. 11th IEEE Inter. Conf. on Computer Vision (ICCV’07), pages 1–8, 2007.