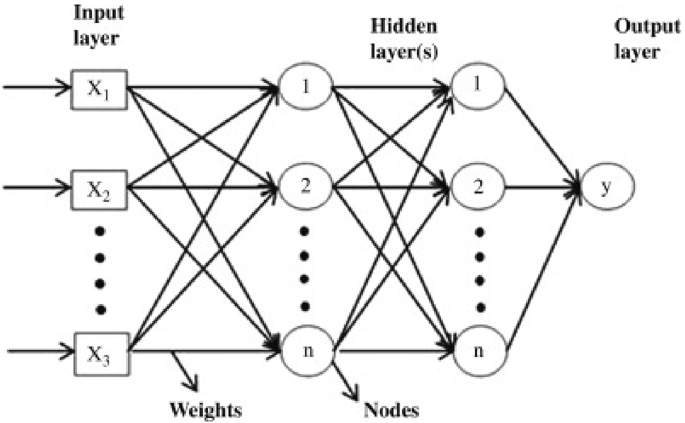
The Use of Matrix Decompositions to Initialize Artificial Neural Networks

Anna Van Boven

University of Puget Sound

May 2, 2021

Neural Networks



Neural Networks

- An Artificial neural network "learns" a training data set so that it can predict the output of other similar data points.
- Supervised learning = labels for data are known. Algorithms "learn" how to label new data.
- Weight matrix holds weights between two layers : $[W]_{ij}$ holds weight of X_i sending to perceptron j.
- Update weights using backpropogation.
- Optimal strategy for initializing weights for a neural network is unknown.

Example: Political Party Prediction

$$X = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \qquad \mathrm{T} = \begin{bmatrix} 0 & 1 & 0 & 1 \end{bmatrix}$$

- Each column represents a politician.
- Each row represents an issue.
- X_{ij} represents how politician j voted on issue i.
- 1 is a yea vote, 0 is a nay vote.
- T_i represents the party affiliation of politician i.
- 1 is Democrat, 0 is Republican.

Singular Value Decompositions to Initialize Weights

Goal

$\min \| T - WX \|_F$

Ideally, $\exists W$ where $WX = T$.
Let $W = T\hat{X}$, where $\hat{X}$ is SVD pseudoinverse of X.
Let the singular value decomposition, $X = U\Sigma V^T$ be rewritten:

$$U = \begin{bmatrix} U_r & U_{m-r} \end{bmatrix} \qquad \Sigma = \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix} \qquad V = \begin{bmatrix} V_r & V_{n-r} \end{bmatrix}$$

Let $W = TV_r\Sigma_r^{-1}U_r^*$. Then, $WX \approx T$.

Political Party Example: SVD

The matrix X has a rank of 3.

$$U = \begin{bmatrix} -0.67 & 0.37 & -0.64 & 0 & 0 \\ -0.45 & -0.52 & 0.17 & -0.61 & -0.36 \\ 0 & 0 & 0 & -0.51 & 0.86 \\ -0.38 & 0.57 & 0.73 & -0 & -0 \\ -0.45 & -0.52 & 0.17 & 0.61 & 0.36 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 2.8 & 0 & 0 & 0 \\ 0 & 1.21 & 0 & 0 \\ 0 & 0 & 0.83 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad V = \begin{bmatrix} -0.38 & 0.77 & 0.1 & 0.5 \\ -0.24 & 0.3 & -0.78 & -0.5 \\ -0.7 & -0.08 & 0.51 & -0.5 \\ -0.56 & -0.55 & -0.36 & 0.5 \end{bmatrix}$$

$$U_r = \begin{bmatrix} -0.67 & 0.37 & -0.64 \\ -0.45 & -0.52 & 0.17 \\ 0 & 0 & 0 \\ -0.38 & 0.57 & 0.73 \\ -0.45 & -0.52 & 0.17 \end{bmatrix} \quad \Sigma_r = \begin{bmatrix} 2.8 & 0 & 0 \\ 0 & 1.21 & 0 \\ 0 & 0 & 0.83 \end{bmatrix} \quad V_r = \begin{bmatrix} -0.38 & 0.77 & 0.1 \\ -0.24 & 0.3 & -0.78 \\ -0.7 & -0.08 & 0.51 \\ -0.56 & -0.55 & -0.36 \end{bmatrix}$$

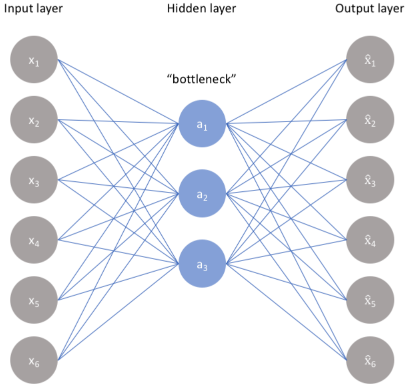
$$W = \begin{bmatrix} 1 & -0 & 0 & -1 & -0 \end{bmatrix}$$

$$WX = \begin{bmatrix} -0 & 1 & -0 & 1 \end{bmatrix}$$

$$T = \begin{bmatrix} 0 & 1 & 0 & 1 \end{bmatrix}$$

Autoencoders

The goal of autoencoders is to reconstruct the original data point.



Non-negative Matrix Factorization (NMF)

- X is a $m \times n$ data matrix with non-negative entries
- $X \approx AS$, $A = m \times p$ matrix, $S = p \times n$ matrix.
  - All entries in A and S are non-negative.
- Update A and S with the following equations:

$$A \leftarrow A \otimes \frac{XS^T}{ASS^T} \qquad S \leftarrow S \otimes \frac{A^T X}{A^T AS}$$

- $\otimes$ denotes entry-wise matrix multiplication.
- Use entry-wise matrix division on the fractions (set denominator values of $0 = 1$).

Non-negative Matrix Factorization (NMF)

If $X \approx AS$, then $\boldsymbol{X}_n \approx A\boldsymbol{S}_n$.

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} = A \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_p \end{bmatrix}$$

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} = \boldsymbol{A}_1 s_1 + \boldsymbol{A}_2 s_2 + \cdots + \boldsymbol{A}_p s_p$$

- A gives basis of new subspace where each column is a basis element.
- S_n gives coordinates of a column X_n for the basis in A.

NMFs to Initialize Weights

Let $f(\boldsymbol{X}_n, W)$ be the approximated output for a single data point.

Goal

$min \sum_{i=0}^{n} \|X_n - f(\boldsymbol{X}_n, W)\|_F$

Let A° be the Moore-Penrose pseudoinverse of A:
$A^\circ = (A^* A)^{-1} A^*$. Then, $S \approx A^\circ X$.

$$\begin{aligned} \|\boldsymbol{X}_n - f(\boldsymbol{X}_n, W)\| &\approx \|\boldsymbol{X}_n - A\boldsymbol{S}_n\| \\ &\approx \|\boldsymbol{X}_n - AA^\circ \boldsymbol{X}_n\| \end{aligned}$$

- The matrix A° is the weight matrix between the input and the bottleneck.
- The matrix AA° is used to approximate the data matrix X.

Initializing NMFs

Choosing value of p:

List the singular values of X in descending order.

Goal

The value p is found such that

$S_p < \alpha$, and $S_{p+1} \geq \alpha$

for $S_m = \frac{\sum_{i=1}^{m} \sigma_i}{\sum_{i=1}^{k} \sigma_i}$.

The value α can be between 0 and 1, normally closer to 1.

Initializing NMFs

Initializing A and S:

Eckhart Young Theorem on Low-Rank Approximation

Let D be an $m \times n$ matrix with singular value decomposition $D = U\Sigma V^T$.

$$U = \begin{bmatrix} U_k & U_{m-k} \end{bmatrix} \qquad \Sigma = \begin{bmatrix} \Sigma_k & 0 \\ 0 & 0 \end{bmatrix} \qquad V = \begin{bmatrix} V_k & V_{n-k} \end{bmatrix}$$

The matrix $\hat{D} = U_k \Sigma_k V_k^T$ solves $min \ \|D - \hat{D}\|_F$.

1. Apply the above theorem to data matrix X with $k = p$.
2. Set $A = |U_p|$.
3. Set $S = |\Sigma_p V_p^T|$.

Political Party Example: Choosing p

$$\sigma = \begin{bmatrix} 2.7986 & 1.2147 & 0.832 & 0 \end{bmatrix} \qquad \sum_{i=1}^{4} \sigma_i = 4.845$$

Let $\alpha = 0.9$.

Goal

Find p where $S_p < 0.9$, and $S_{p+1} >= 0.9$.

p	S_p	S_{p+1}
1	.577	.828
2	.828	1

With $\alpha = 0.9$, the table shows $p = 2$.

Political Party Example: Initializing A and S

$$U_p = \begin{bmatrix} -0.67 & 0.37 \\ -0.45 & -0.52 \\ 0 & 0 \\ -0.38 & 0.57 \\ -0.45 & -0.52 \end{bmatrix} \qquad \Sigma_p = \begin{bmatrix} 2.8 & 0.0 \\ 0 & 1.21 \end{bmatrix} \quad V_p = \begin{bmatrix} -0.38 & 0.77 \\ -0.24 & 0.3 \\ -0.7 & -0.08 \\ -0.56 & -0.55 \end{bmatrix}$$

$$A_i = \begin{bmatrix} 0.67 & 0.37 \\ 0.45 & 0.52 \\ 0 & 0 \\ 0.38 & 0.57 \\ 0.45 & 0.52 \end{bmatrix} \qquad S_i = \begin{bmatrix} 1.05 & 0.67 & 1.95 & 1.57 \\ 0.94 & 0.37 & 0.1 & 0.67 \end{bmatrix}$$

$$A = \begin{bmatrix} 0.36 & 0.78 \\ 0.45 & 0 \\ 0 & 0 \\ 1.49 & 0.6 \\ 0.45 & 0.0 \end{bmatrix} \qquad S = \begin{bmatrix} 0.05 & 0.14 & 2.19 & 2.25 \\ 1.4 & 0.75 & 0.54 & 0.17 \end{bmatrix}$$

Political Party Example: NMFs

$$X = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \qquad AA^{\circ}X = \begin{bmatrix} 1.03 & 0.88 & 0.63 & 0.48 \\ 0.06 & -0.2 & 0.36 & 0.11 \\ 0 & 0 & 0 & 0 \\ 0.96 & 0.15 & 1.47 & 0.66 \\ 0.06 & -0.2 & 0.36 & 0.11 \end{bmatrix}$$

$$S = \begin{bmatrix} 0.05 & 0.14 & 2.19 & 2.25 \\ 1.4 & 0.75 & 0.54 & 0.17 \end{bmatrix} \quad A^{\circ}X = \begin{bmatrix} 0.13 & -0.45 & 0.81 & 0.24 \\ 1.27 & 1.35 & 0.44 & 0.51 \end{bmatrix}$$

Analysis

- Most common technique is random initialization:
  - No technique discussed can put a bound on distance between initialization and optimal weights
  - Computing SVD has a runtime $O(min(mn^2, m^n))$. Could slow down runtime of weight computation.
- SVD technique can only be used on single layer neural network.
- NMF technique only works on non-negative data.

Sources

- Atif, Syed Muhammad, et al. "Improved SVD-Based Initialization for Nonnegative Matrix Factorization Using Low-Rank Correction." ScienceDirect, vol. 122, 1 May 2019.
- Barata, J. C. A., and M. S. Hussein. "The Moore-Penrose Pseudoinverse. A Tutorial Review of the Theory." Cornell University, 31 Oct. 2011.
- Flenner, Jennifer, and Blake Hunter. "A Deep Non-Negative Matrix Factorization Neural Network." Claremont Mckenna College, 2016.
- Jordan, Jeremy. "Introduction to Autoencoders." Jeremyjordan, 19 Mar. 2018, www.jeremyjordan.me/autoencoders/.
- Schafer, Casey. "The Neural Network, Its Techniques and Applications." Whitman University, 12 Apr. 2016.
- Lee, D., Seung, H. Learning the parts of objects by non-negative matrix factorization. Nature 401, 788–791 (1999). https://doi.org/10.1038/44565
- Gillis, N. (2014). The why and how of nonnegative matrix factorization. Regularization, Optimization, Kernels, and Support Vector Machines, 12, 257–291.
- Gillis, N., amp; Glinuer, F. (2012). A multilevel approach for nonnegative matrix factorization. ScienceDirect, 236(7). Retrieved 2021, from https://www.sciencedirect.com/science/article/pii/S0377042711005334
- Nagyfi, R. (2018, September 4). The differences between Artificial and Biological Neural Networks [Web log post]. Retrieved from https://towardsdatascience.com/the-differences-between-artificial-and-biological-neural-networks-a8b46db828b7
- Qiao, H. (2014, October 10). New SVD based initialization strategy for Non-negative Matrix Factorization [PDF]. Ithaca, NY: Cornell University.
- Squires, S., Prugel-Bennett, A., Miranjan, M. (2017). Rank Selection in Nonnegative Matrix Factorization using Minimum Description Length [PDF]. Southampton, UK: University of Southampton.
- Https://medium.com/@datamonsters/artificial-neural-networks-in-natural-language-processing-bcf62aa9151a [Web log post]. (2017, August 17). Retrieved 2021, from Applications of Artificial Neural Networks in Natural Language Processing