Linear Algebra for Machine Learning

TN Interlude

Disclaimer: Work in progress. Portions of these written materials are incomplete.

Tensor Networks

An efficient way of storing and manipulating high-dimensional complex data using linear algebraic techniques.

Tensor Networks

A fun/easy/insightful diagrammatic way of doing linear algebra!

Tensor Networks

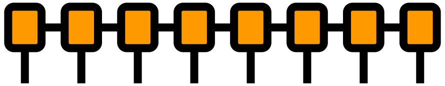
- Data structures and algorithmic tools with origins in quantum physics.

- Tool for optimization of high dimensional data by generalizing matrix multiplication to tensor network contraction.

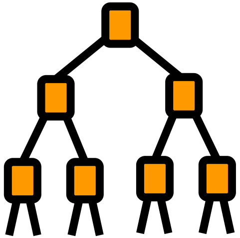
- Many fundamental applications already in the sciences, with emerging use cases in machine learning.

What is a tensor network?

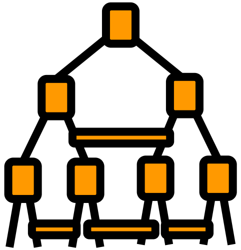
tensor network = data structure to **efficiently encode** a **large tensor** and whose properties are summarized by **diagrams like these**



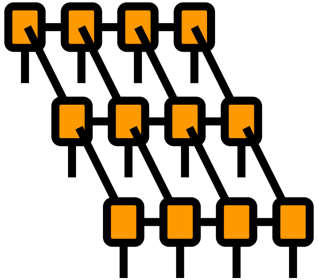
matrix product state
(MPS)



tree tensor network
(TTN)



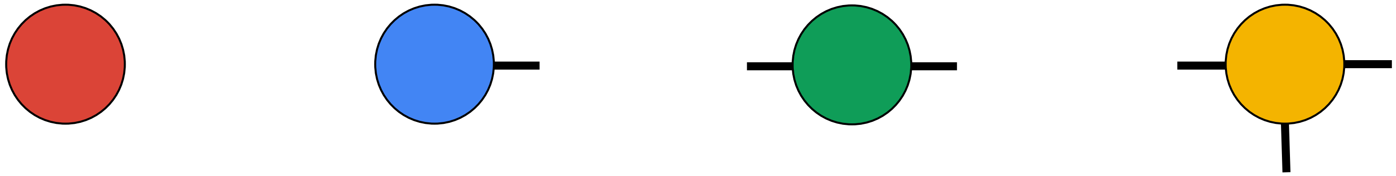
Multi-scale entanglement
renormalization ansatz
(MERA)



Projected entangled
pair states
(PEPS)

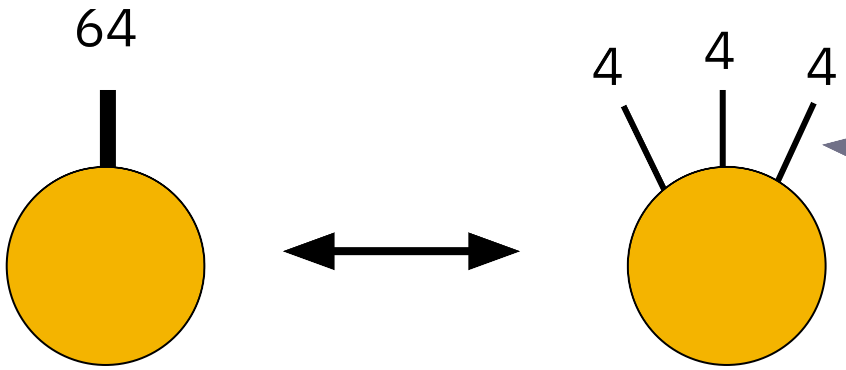
Refresher on tensors

Scalar	Vector	Matrix	3-Tensor
$\begin{bmatrix} 1.32 \end{bmatrix}$	$\begin{bmatrix} 1.32 \\ 0.92 \end{bmatrix}$	$\begin{bmatrix} 1.32 & 3.65 \\ 0.92 & 0.01 \end{bmatrix}$	$\begin{bmatrix} 9.01 & 2.56 \\ 7.71 & 0.21 \end{bmatrix}$ $\begin{bmatrix} 1.32 & 3.65 \\ 0.92 & 0.01 \end{bmatrix}$



Number of parameters	1	2	4	8

Tensors can be reshaped



Legs can have any dimension! Not just 2 or 4 or 64.

reshaping = split / combine edges

ADVANCED VIEW!
Can think of this in terms of a chosen tensor decomposition of vector spaces:

$$V = V_1 \otimes V_2 \otimes V_3$$

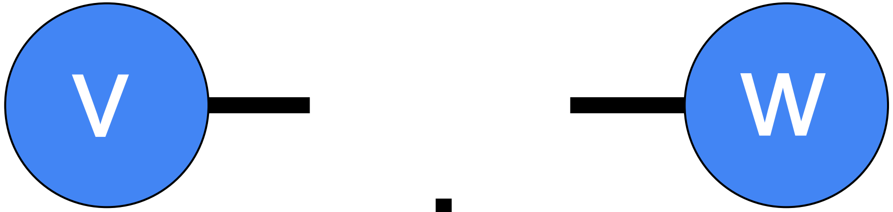
Diagram for tensor products

$$v \otimes w$$

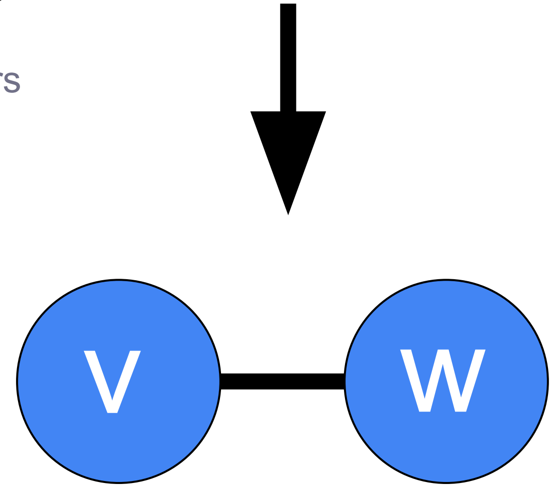
$$v_i w_j$$

Treating these two vectors as a single two-leg diagram corresponds to the tensor product of the vectors

Tensor Contraction



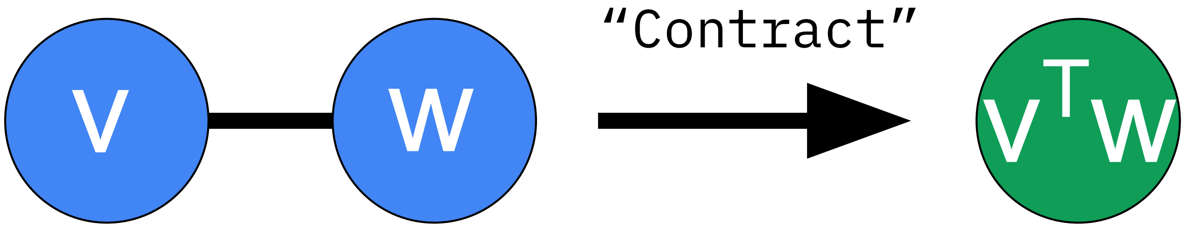
Stitching together tensors corresponds to “dot-like” operations.



WARNING! It only makes sense to stitch together legs that have the same dimension! The notation does not necessarily reflect that information!

No dangling legs left over in this case means the result is a scalar.

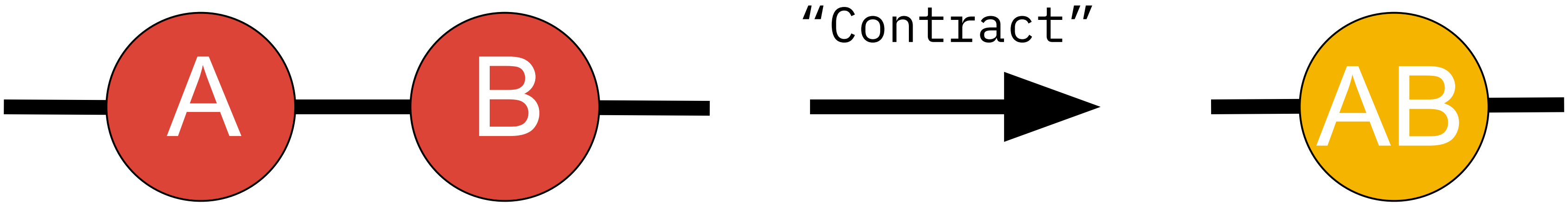
Diagrammatic notation for scalar product



`vector * vector -> scalar`

$$\sum_i v_i w_i$$

Diagrammatic notation for matrix multiplication

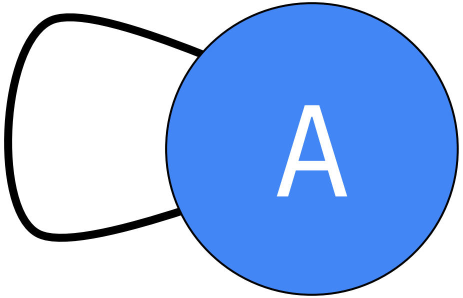


`matrix * matrix -> matrix`

$$\sum_j A_{ij} B_{jk}$$

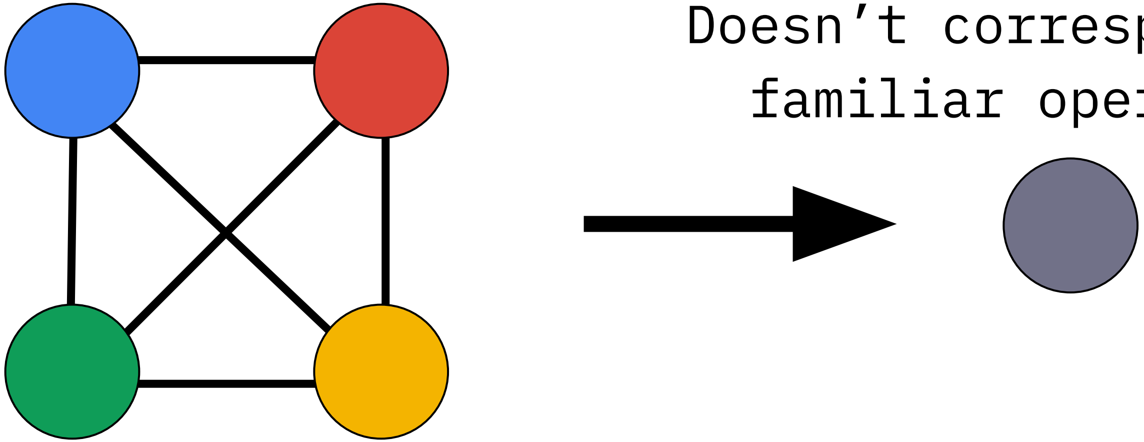
Trace of a matrix

A matrix with its two dangling legs stitched together.



$$\mathrm{tr}(A) = \sum_i A_{ii}$$

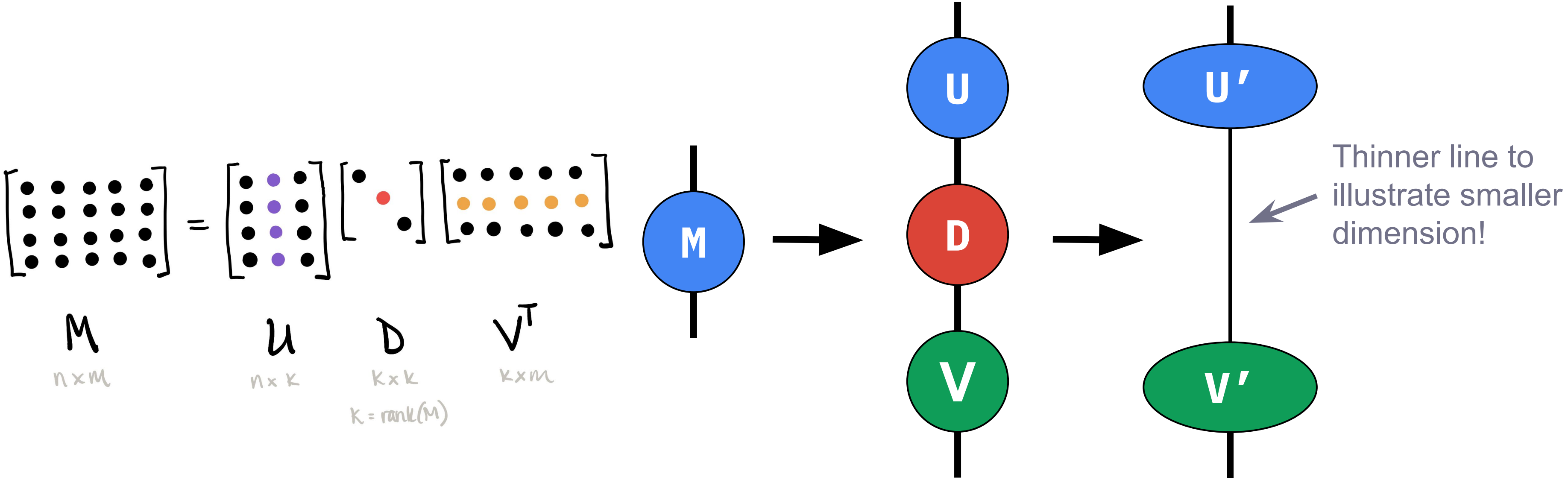
More general contraction



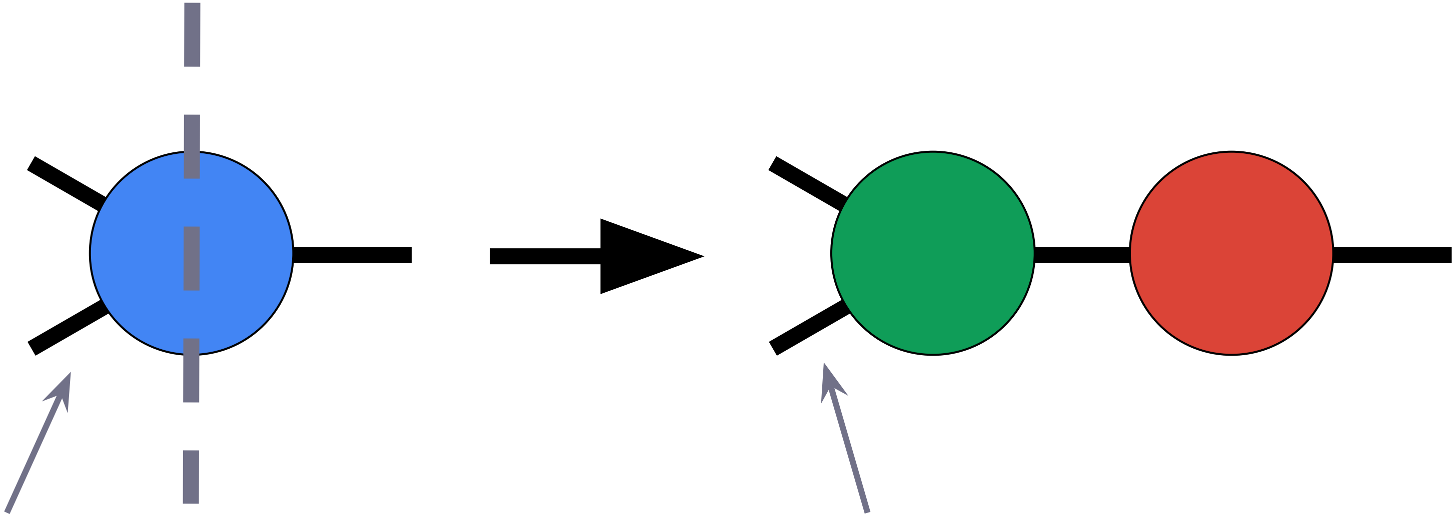
Doesn't correspond to a familiar operation!

$$\sum_{i,j,k,l,m,n} A_{ijk} B_{klm} C_{mjn} D_{nli}$$

SVD has a diagram too!



Reshape + SVD = Split any node!

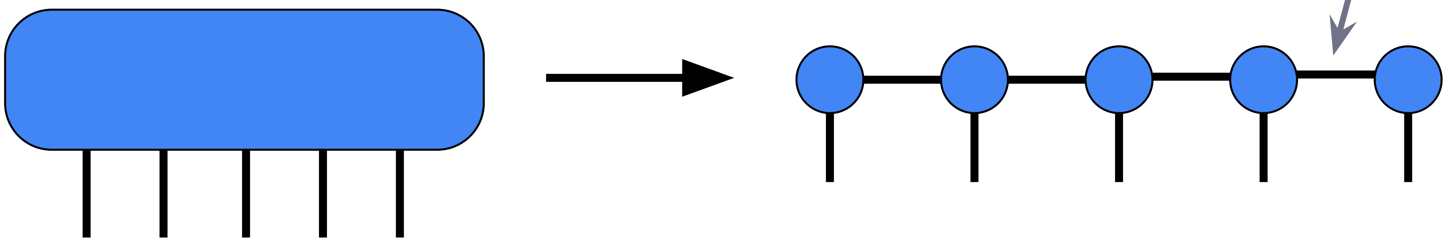


Imagine combining these two legs into a single "thicker" leg and then doing SVD on the resulting matrix.

After the SVD, the legs can be separated again.

Matrix Product State (TN Example)

efficiently encode a large tensor



Taking "bond dimension" to be p

2^N

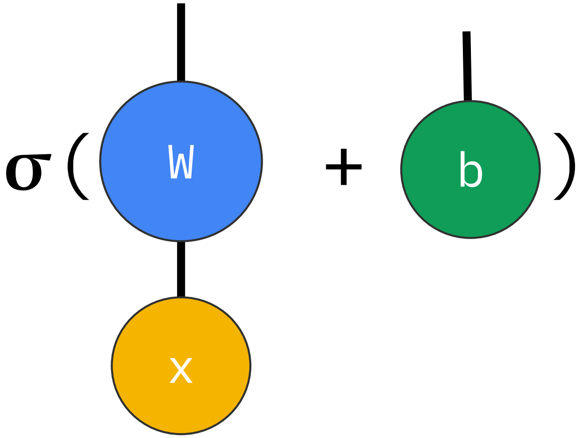
$2 * p^2 * N$

For N large

TN Layers in ML

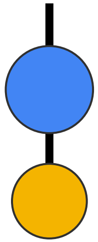
A neural network layer in TN language

$\sigma(Wx + b) ==$

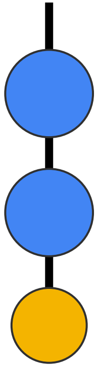


Examples of TN Layers

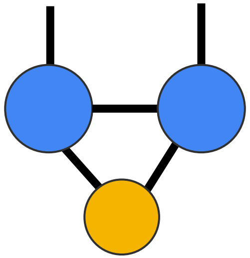
original

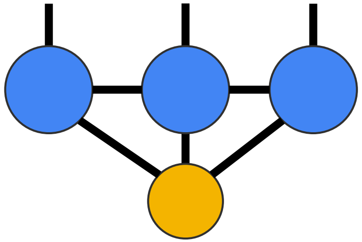


simple
decomposition
(like SVD)

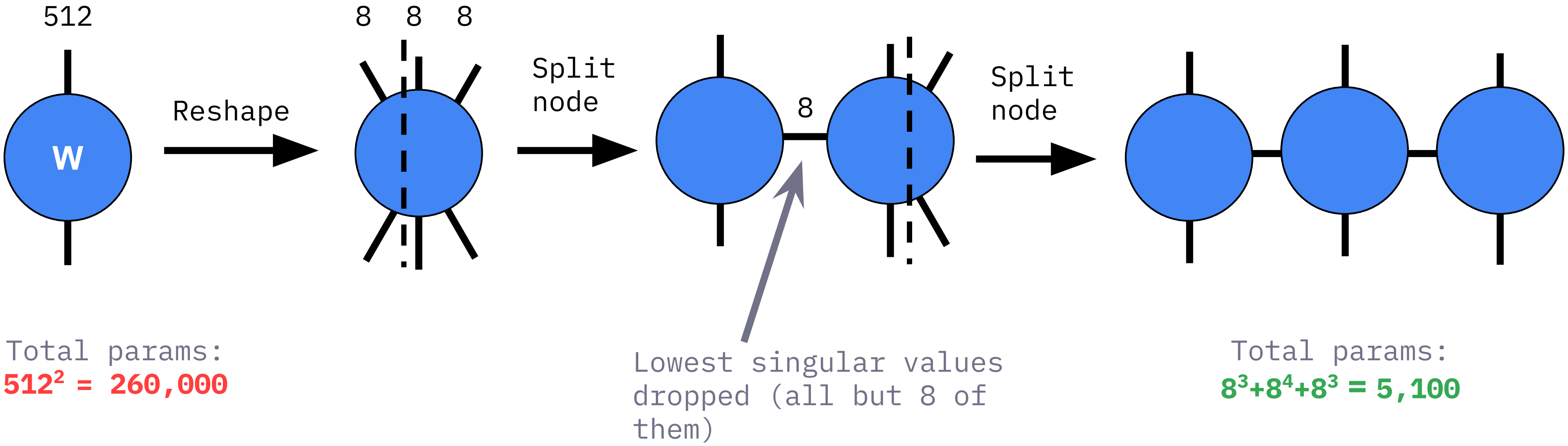


2- and 3-node MPO





Decomposition and compression of existing layers

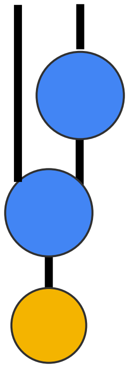


Total params:
512^2 = 260,000

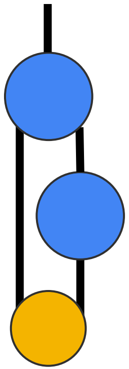
Total params:
$8^3+8^4+8^3$ = 5,100

Other kinds of TN Layers

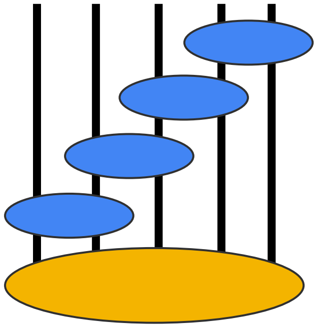
Up Projection



Down Projection



Entangler



Open Source Layers

github.com/google/tensornetwork

Anomaly Detection with TNs

2006.02516

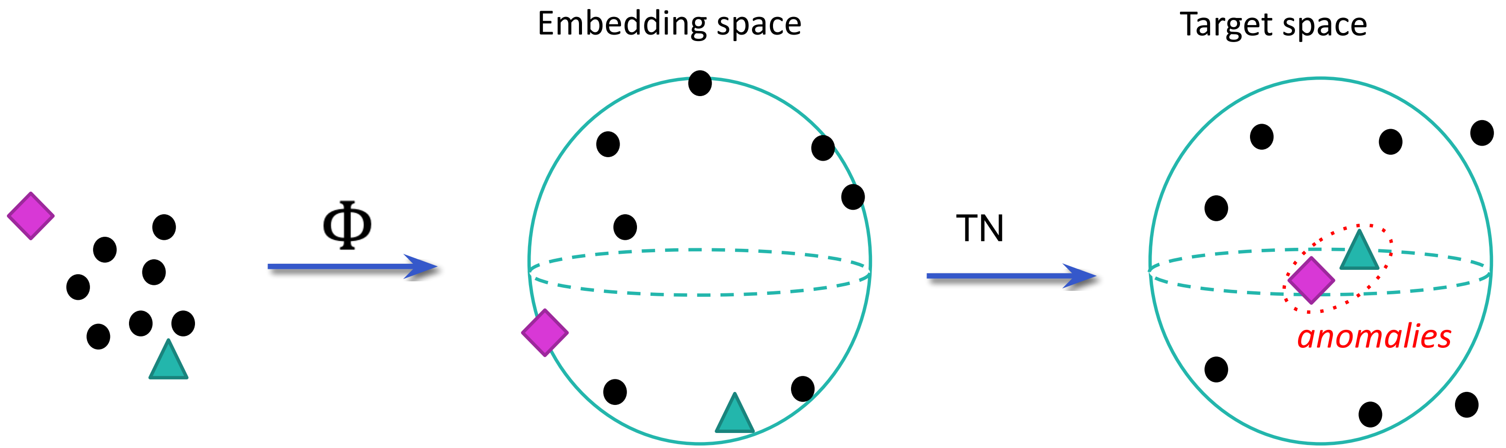
Linear model on an exponentially-large space

We use a linear model for anomaly detection.

To get something nontrivial, we use a linear model based on an exponentially-large space.

Example: Each distinct black-and-white image is taken to be linearly independent in our space.

Model Overview



$$\mathcal{L}_{batch} = \frac{1}{B}\sum_{i=1}^{B}\left(\log \|P\Phi(\boldsymbol{x}_i)\|_2^2 - 1\right)^2 + \alpha \mathrm{ReLU}(\log \|P\|_F)$$

Log for numerical stability

Regulator coefficient