

Manifold Learning

Javier Carrillo Martínez, Centro de Investigación en Matemáticas. Unidad Monterrey
Email: javier.carrillo@cimat.mx

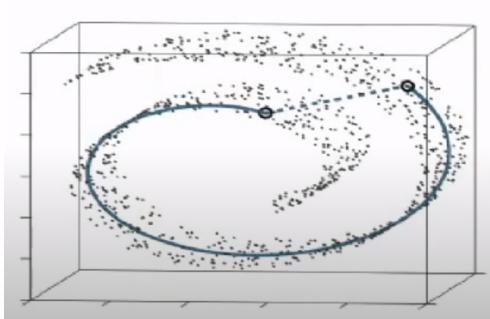


Figura 1. Conjunto de datos con forma de Swiss Roll como muestra de estructuras no lineales.

Resumen—Se presentan tres métodos de reducción de dimensiones no lineal (ISOMAP, LLE y t-SNE), en el contexto de Manifold Learning, el desempeño de los métodos se mide probando sobre distintos conjuntos de datos, obteniendo una mejor clustrización con el método de t-SNE. Por último se muestra una propuesta para ubicar datos nuevos del conjunto de prueba en un mapa ya creado t-SNE usando redes neuronales.

I. INTRODUCCIÓN

Trabajar con datos cuya dimensionalidad es alta es un problema complejo que motiva muchos problemas en distintas áreas de las ciencias computacionales y la estadística, donde los esfuerzos se reparten en crear técnicas para ajustar modelos a este tipo de conjuntos, visualizarlo y hacer transformaciones para tener representaciones de menor dimensión que es el caso del presente trabajo.

Dentro de los trabajos realizados para hacer técnicas de reducción de dimensiones se encuentran las reducciones de dimensión lineales como PCA, LDA y MDS, que son técnicas del siglo pasado, que funcionan muy bien para datos que se pueden representar en una subespacio lineal de menor dimensión, lo cuál puede aplicar a muchos conjuntos de datos, sin embargo presenta conflictos a la hora de estudiar conjuntos de datos que presentan una forma claramente no lineal como la que se muestra en la figura 1 en el famoso ejemplo del Swiss Roll.

El primer concepto importante a presentar es el concepto de variedad o manifold en inglés, que podemos pensarlo como la generalización de una superficie en un espacio d -dimensional, una variedad puede entonces representar un sin fin de figuras m -dimensionales ($d \leq m$) con formas no lineales. En general si suponemos que los datos m -dimensionales recaen sobre una variedad d -dimensional incrustada en el espacio m

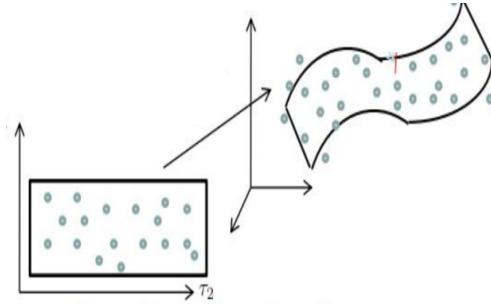


Figura 2. Representación esquemática de la transformación de una variedad de un espacio superior a una superficie en un espacio bidimensional.

dimensional con $d \leq m$, es posible construir un espacio de menor dimensión a partir de una transformación, como se ilustra en la figura 2.

Los métodos de Manifold Learning consisten en explotar este tipo de estructuras m -dimensionales en la que los datos se encuentran embedidos y pretenden recuperar información de esta estructura para crear representaciones de menor dimensión, para esto se realizan un par de supuestos que podemos resumir como:

- Los datos tienen representación vectorial con estructura (recaen sobre una variedad).
- Los datos son lo suficientemente cercanos y distribuidos uniformemente.
- Poco ruido.

Estos supuestos nos ayudan como indicio de cuándo se pueden usar este tipo de métodos. En el presente trabajo se introducirán de forma breve y con un par de ejemplos tres de los métodos más usados actualmente para reducción de dimensiones no lineal, en particular en el contexto de Manifold Learning.

II. MÉTODOS

En esta sección se presentan 3 métodos de reducción de dimensiones clásicos del área de Manifold learning, se presentan las ideas básicas detrás de los métodos: mapas isométricos y locally linear embedding, mientras que se presenta de forma ligeramente más extensa la formulación sobre el método t-distributed Stochastic Neighbor Embedding y a modo de comparación se presenta el desempeño de los tres métodos mediante tres distintas bases de datos.

Los conjuntos de datos corresponden, primero a la base de datos escritos a mano y digitalizados MNIST que consta de 70,000 observaciones con 784 columnas para cada uno, la segunda base de datos consiste en un set de 1300 fotos de 13 frutas distintas y por último se utilizó un conjunto de reseñas sobre artículos de 8 categorías distintas y con un sentimiento asociado a cada reseña, esta base de datos representa un ejemplo de información no vectorial que recibió un preprocessamiento eliminando caractéres raros y haciendo bag of words para posteriormente realizar una vectorización de estos resultados.

II-A. Mapa isométrico

La idea básica del mapa isométrico es suponer que los datos forman una variedad suave que se puede reconstruir de manera aproximada para calcular igualmente de manera aproximada las geodesias entre cada dato, obteniendo así una matriz de distancias con las que se puede realizar una representación de menor dimensión mediante escalamiento multidimensional.

De modo que las tres etapas de las que el método está comprendido se pueden identificar como:

- Reconstruye la variedad mediante un grafo de vecinos usando.
- Estima las distancias sobre la variedad, mediante las distancias más cortas sobre el grafo.
- Construye el mapa d-dimensional usando MDS sobre las distancias de la fase 2.

Para realizar la primer fase se requiere de la elección de los vecinos a tomar en cuenta y la creación de un grafo, para la elección de los vecinos se realiza el método de k-vecinos más cercanos para cada observación, donde el número de vecinos es un parámetro del modelo, y con los vecinos cercanos se crea una matriz de adyacencias que representa un grafo no dirigido que aproxima la variedad formada por los datos. En la paquetería de Sklearn en Python, este procesos se realiza mediante el algoritmo Ball Tree que tiene complejidad $O[D \log(k)N \log(N)]$, con D el número de dimensiones del espacio de dimensión alta, k el número de vecinos y N el número de puntos.

Realizar la segunda fase es quizás la más complicada y la que demanda de una capacidad computacional mayor, pues se requiere crear una matriz de distancias a partir de las distancias mínimas entre cada par de puntos a través del grafo, el mejor algoritmo para determinar la menor distancia entre dos puntos en un grafo no dirigido es el algoritmo de Djikstra's con complejidad $O[N^2(k + \log(N))]$.

Por último en la tercera fase del método, la realización de la representación de baja dimensión se realiza mediante un MDS clásico, que se reduce al cálculo de eigenvalores que en el método implementado por Sklearn es realizada con complejidad $O[dN^2]$ con d el

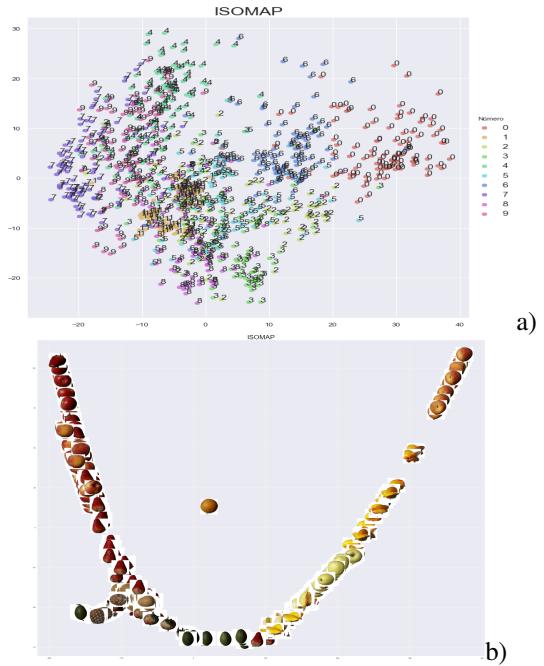


Figura 3. Método de clusterización ISOMAP probado en 2 conjuntos de datos, a) conjunto de 70,000 dígitos escritos a mano y digitalizados MNIST, b) conjunto de 100 fotografías sobre 13 frutas distintas en distintas etapas de maduración.

número de dimensiones de la representación de baja dimensión, de modo que la complejidad total del método es $O[D \log(k)N \log(N)] + O[N^2(k + \log(N))] + O[dN^2]$.

Para evaluar el desempeño del método se probó sobre los tres conjuntos de datos descritos anteriormente, los resultados se muestran en las figuras 3 para los primeros dos y 4, para el tercero, para los primeros dos métodos se puede observar que la separación de clusters es relativamente buena, en el caso de 3 a) se pueden apreciar agrupados cada dígito, donde los que presentan mayor confusión son los cuatros, nueves y unos, que de antemano sabemos que son ligeramente similares en morfología.

Por otro lado, la figura 3 b) presenta un comportamiento similar en el sentido de que permite separar bien un par de clases del conjunto, en este caso los duraznos y quizás las carambolas, sin embargo hay varias frutas que se muestran bastante mezcladas por lo que no parecen muy confiable usar esta técnica para este conjunto de datos.

La figura 4 nos muestra que aplicar esta técnica sobre los datos de las reseñas da un resultado interesante, primero se puede observar que en la figura de la izquierda a) se muestran dos clusters asociados a cada etiqueta lo cual cobra sentido al recordar que además hay dos tipos de sentimientos asociados y observando la figura de la derecha b) se observa que en efecto una de las repeticiones está asociada a cada sentimiento, este fenómeno se repite para todos los métodos,

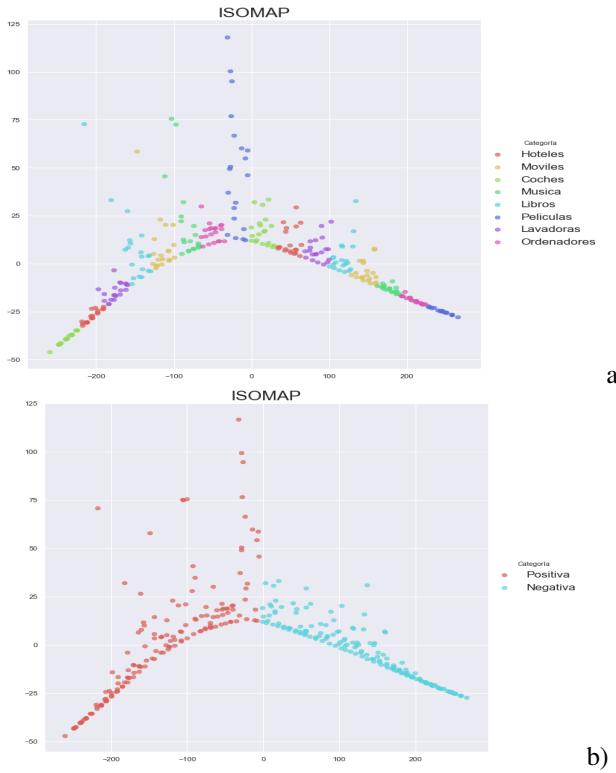


Figura 4. Método de clusterización ISOMAP probado en un conjunto de datos sobre 400 reseñas acerca de 8 artículos con un sentimiento asociado, positivo o negativo, con etiquetas sobre la categoría de la reseña a) y b) el sentimiento asociado.

de modo que en el futuro sólo se muestra la gráfica asociada a las etiquetas de categoría.

Observando un poco más esta figura se puede percibir que la separación es relativamente buena, pues no nos muestra los conjuntos mezclados, sin embargo identificar los clusters sería sumamente complicado si no se tuviera la ayuda de las etiquetas.

II-B. Incrustamiento localmente lineal

Al igual que en el caso del método anterior, el incrustamiento localmente lineal, LLE por sus siglas en inglés, supone que los datos forman una variedad suave que se puede caracterizar localmente mediante la elección de vecinos al rededor de un punto y realizando un análisis muy similar al realizado en componentes principales que finalmente se comparan globalmente para encontrar la mejor representación no lineal.

Al igual que en ISOMAP se puede resumir el método en tres etapas:

- Asignar vecinos a cada punto X_i .
- Calcular la matriz de pesos que mejor reconstruyan linealmente cada punto a partir de sus vecinos.
- Calcular los vectores en menor dimensión mejor reconstruidos por los pesos minimizando la función de

costo.

La primer fase es prácticamente análoga a ISOMAP en el sentido de realizar una elección de vecinos a tomar en cuenta en la etapa siguiente, en la paquetería de Sklearn en Python, este proceso se realiza mediante K-vecinos más cercanos cuya complejidad es $O[D \log(k)N \log(N)]$, con D el número de dimensiones del espacio de dimensión alta, k el número de vecinos y N el número de puntos.

Para la segunda fase suponemos que los datos son locales linealmente, en particular pensaremos que un punto y sus vecinos viven en un subespacio lineal, de modo que el punto en cuestión x_i se puede expresar como una combinación lineal de sus vecinos, es decir:

$$\mathbf{x}_i = \sum_j w_{ij} \mathbf{x}_j,$$

lo cual se puede plantear como un problema de mínimos cuadrados, es decir queremos encontrar los pesos w_{ij} que minimicen la expresión

$$\|\mathbf{x}_i - \sum_j w_{ij} \mathbf{x}_j\|^2$$

donde los $w_{ij} = 0$ para los puntos \mathbf{x}_j que no son vecinos de x_i , esto requiere la solución de un sistema de ecuaciones de $k \times k$ para cada uno de los N vecindarios, de modo que la complejidad está dada por $O[DNk^3]$.

Por último, la fase 3 consiste en encontrar los vectores y_i en el espacio de menor dimensión mediante la minimización de la función de costo que al final se reduce a encontrar los eigenvectores asociados a los eigenvalores de menor magnitud de la expresión

$$\mathbf{Y}^T \mathbf{M} \mathbf{Y} \text{ con } \mathbf{M} = (\mathbf{I} - \mathbf{W})^T (\mathbf{I} - \mathbf{W}).$$

De modo que esta fase tiene complejidad $O[dN^2]$, por lo que la complejidad total del método está dada por $O[D \log(k)N \log(N)] + O[N^2(k + \log(N))] + O[dN^2]$.

Al igual que el caso anterior se probó el método sobre los tres conjuntos de datos, los resultados se muestran en las figuras 5 y 6, a diferencia de lo observado con el método ISOMAP en las figuras a) y b) podemos observar que la mayoría de los datos están mezclados y no se puede observar una separación propiamente de las clases.

En el caso de la prueba con los datos de reseñas se pueden observar un comportamiento similar al observado con ISOMAP donde las clases no se encuentran tan mezcladas, pero no se percibe mucha separación entre ellas, de modo que en general este método no parece ser adecuado para ser utilizado con estos conjuntos de datos.

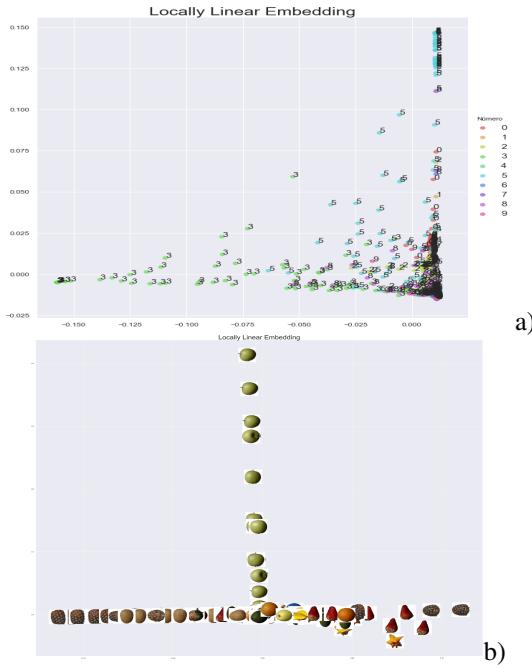


Figura 5. Método de clusterización LLE probado en 2 conjuntos de datos, a) conjunto de 70,000 dígitos escritos a mano y digitalizados MNIST, b) conjunto de 100 fotografías sobre 13 frutas distintas en distintas etapas de maduración.

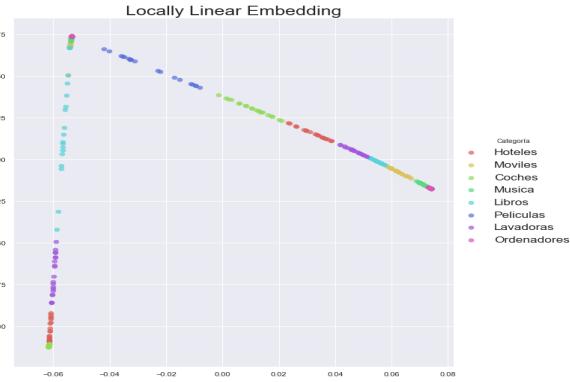


Figura 6. Método de clusterización LLE probado en un conjunto de datos sobre 400 reseñas acerca de 8 artículos con un sentimiento asociado, positivo o negativo, con etiquetas sobre la categoría de la reseña a) y b) el sentimiento asociado.

III. T-DISTRIBUTUITED STOCHASTIC NEIGHBOOR EMBEDDING

El método t-SNE por sus siglas en inglés es distinto en su planteamiento respecto a los dos métodos anteriores, pues no parte de la idea de recuperar la estructura de la variedad en la que se encuentran embebidos los datos ni de calcular distancias entre los datos, sino que mide la similitud de estos mediante probabilidades y realiza un proceso de aprendizaje mediante la minimización de una cantidad que compara estas similitudes en el espacio de dimensión alta y las compara con las obtenidas en el espacio de dimensión baja.

Para esto imaginemos que fijamos un punto x_i y asignamos una distribución gaussiana centrada en este punto, de modo que podemos medir la similitud de otros puntos x_j mediante la probabilidad condicional dada por:

$$p_{ij} = \frac{\exp\left(-\|x_i - x_j\|^2 / 2\sigma^2\right)}{\sum_{k \neq i} \exp\left(-\|x_k - x_i\|^2 / 2\sigma^2\right)},$$

aquí el parámetro σ nos permite regular la cercanía a partir de la cuál consideraremos los puntos como vecinos, de manera similar al parámetro k sobre el número de vecinos a considerar en los métodos anteriores.

De forma análoga se mide la similitud entre los puntos en el espacio de dimensión baja, con la excepción de usar una distribución Gaussiana y usando en su lugar una distribución t-Student con un grado de libertad que debido a sus colas pesadas permite obtener valores más grandes en la asignación de vecinos en el espacio de baja dimensión, de modo que la probabilidad está dada por la expresión:

$$q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k \neq i} (1 + \|y_k - y_i\|^2)^{-1}}.$$

Ahora, para comparar estas similitudes y plantear un proceso de aprendizaje planteamos como función de costo a la cantidad conocida como divergencia de Kullback-Leibler que nos mide de forma no simétrica la similitud entre dos funciones de probabilidad y está dada por:

$$C = KL(P\|Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}},$$

decimos que es una medida de similitud asimétrica pues al observar la forma de la ecuación podemos notar que al tener valores altos de p_{ij} y bajos de q_{ij} la penalización es mayor que si tenemos valores bajos de p_{ij} y altos de q_{ij} , esto quiere decir que a nuestro proceso de aprendizaje buscará disminuir la asignación de dos puntos alejados en el espacio de baja dimensión cuando los puntos en el espacio de dimensión alta eran cercanos, que el caso contrario.

Ahora, para resolver nuestro problema debemos realizar la minimización de la función de costo en función de los vectores que determinan la representación de baja dimensión, de modo que derivando respecto a los valores de y_i se obtiene la siguiente expresión:

$$\frac{\delta C}{\delta y_i} = 4 \sum_j (p_{ij} - q_{ij}) (y_i - y_j) \left(1 + \|y_i - y_j\|^2\right)^{-1},$$

que puede ser pensada como una fuerza de tipo resorte ejercida por todos los puntos y_j sobre el punto y_i , lo cual representa un gran alto computacional, sin embargo el autor del algoritmo [3] reconoció que este problema es análogo

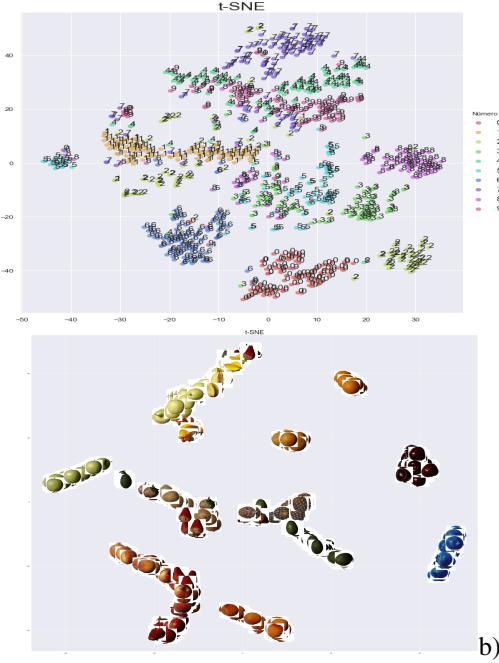


Figura 7. Método de clusterización t-SNE probado en 2 conjuntos de datos, a) conjunto de 70,000 dígitos escritos a mano y digitalizados MNIST, b) conjunto de 100 fotografías sobre 13 frutas distintas en distintas etapas de maduración.

al problema clásico de los N cuerpos que pretende resolver la dinámica de un conjunto de N partículas generada por su interacción, por lo que acude a la aproximación de Barnes-Hut usada en Astronomía para resolver este problema para calcular las órbitas de un cuerpo celeste debido a la presencia de otros cuerpos celestes.

La aproximación de Barnes-Hut consiste en realizar pequeños clusters de puntos y aproximar la fuerza sentida por el punto x_i debido a los puntos inmiscuidos en el cluster mediante la fuerza debida debido a su centro de masa escalada por el número de puntos en el cluster, con esta aproximación se logra hacer que el algoritmo sea más eficiente y disminuye el tiempo de cómputo.

La paquetería Sklearn de Python 3 tiene una implementación de este método la cual se utilizó para calcular dos componentes tomando como parámetros los default de éste, que son 30 como número de perplexidad, 200 como taza de aprendizaje del descenso del gradiente y 1000 para el máximo de iteraciones para realizar los mapas mostrados en las figuras 7 y 8.

Al observar la figura 7 lo primero que podemos observar es que los mapas lucen bastante segmentados y las clases tienden a agruparse, lo cual pone a este método por encima de los dos anteriores pues en caso de no contar con las etiquetas sobre las clases este método nos ayudaría a observar más clusters y hacer una caracterización posterior sobre las

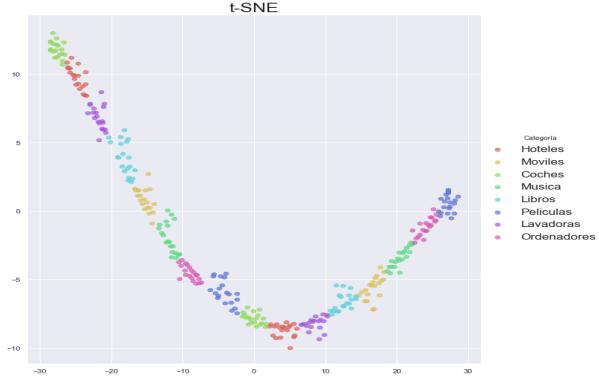


Figura 8. Método de clusterización t-SNE probado en un conjunto de datos sobre 400 reseñas acerca de 8 artículos con un sentimiento asociado, positivo o negativo, con etiquetas sobre la categoría de la reseña a) y b) el sentimiento asociado.

características de cada cluster. En el caso de los datos del MNIST a) como siempre podemos observar que la zona más conflictiva es

Por otro lado, la figura 8 no presenta una mejor considerable respecto a los dos métodos anteriores pues si bien las clases no están mezcladas, no muestran separación para poder ser apreciadas como clusters.

III-A. Representando nuevos datos

Una pregunta usual al obtener este tipo de representaciones, es si se pueden representar nuevos datos en el mapa obtenido sin tener que volver a realizar el algoritmo sobre todos los datos, lo cuál en principio no se puede realizar debido a que ninguno de los métodos presentados realiza un aprendizaje explícito de la función que realiza la parametrización entre ambos espacios, sin embargo Van der Maaten propone dos opciones [4] uno es una implementación sofisticada de un regresor sobre las pérdidas del algoritmo usando redes neuronales y maquinas de Boltzmann restringidas que él implementa en la referencia [5], la otra opción es bastante más simple y consiste en entrenar un regresor sobre el mapa a partir de los datos de la dimensión superior.

En las figuras mostradas en 9 se muestran los resultados de implementar esta idea mediante un perceptrón multicapa con capa de entrada de 100 nodos, una capa escondida, función de activación Relu y 500 como número máximo de iteraciones, sobre los primeros dos conjuntos de datos.

En el caso de los datos del MNIST se entrenó la red sobre el mapa obtenido con t-SNE con 10,000 puntos y se muestra la predicción de 1000 puntos de un conjunto de prueba en la figura 9a) donde se observa que la representación es bastante buena, pues se observan las mismas características observadas en el mapa obtenido de ligera separación y buena agrupación de las clases con confusión sobre los 4s y 9s.

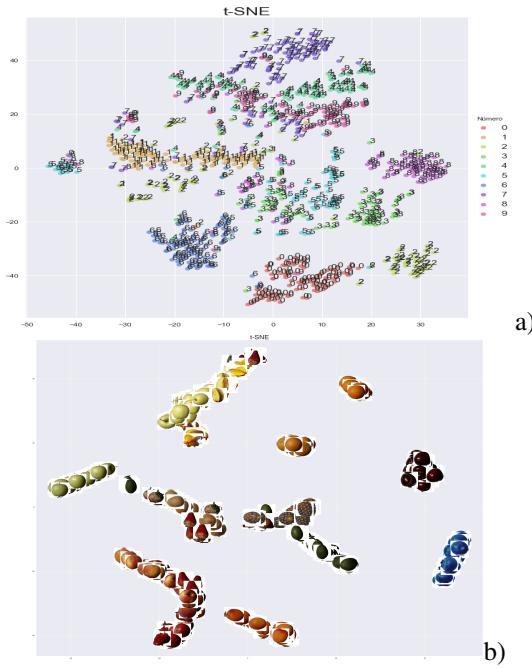


Figura 9. Predicción de un mapa obtenido a partir de un regresor de perceptrón multi capa entrenado con un mapa t-SNE en 2 conjuntos de datos, a) conjunto de 70.000 dígitos escritos a mano y digitalizados MNIST, b) conjunto de 100 fotografías sobre 13 frutas distintas en distintas etapas de maduración.

Por otro lado, al realizar el ejercicio sobre los datos de frutas en la figura 9b) se observa una buena clusterización de los datos, con los grupos separados y clases agrupadas y las posiciones de las frutas se encuentran en posiciones similares a las mostradas en la figura 7b), de modo que parece que este método de regresión funciona bastante bien para predecir la posición de los puntos en el mapa.

IV. CONCLUSIONES

Tras realizar un breve comparativo sobre los métodos ISOMAP, LLE y t-SNE se pudo observar que el mejor desempeño en el sentido de mejores representaciones obtenidas, de los métodos sobre estos conjuntos de datos se alcanzó utilizando el método t-SNE, a excepción de los datos de reseñas que muestran un comportamiento bastante similar en todos los métodos.

Se mostró la viabilidad de entrenar un modelo de regresión con redes neuronales para representar puntos nuevos sobre un mapa t-SNE ya obtenido, dando paso así a realizar este método de forma más robusta.

REFERENCIAS

- [1] B. Tenenbaum Josua, De Silva Vin Langford Jhon, *A Global Geometric Framework for Nonlinear Dimensionality Reduction*, Science 290, 2000.
- [2] T. Roweis Sam K. Saul Lawrence, *Nonlinear Dimensionality Reduction by Locally Linear Embedding*, Science 290, 2000.
- [3] Van der Maaten Hinton Geoffrey, *Visualizing Data using t-SNE*, Journal of Machine Learning Research 9, 2008.
- [4] Van der Maaten, *FAQ about t-SNE*, recuperado del Github del autor el 10 de Junio de 2020 <https://lvdmaaten.github.io/>
- [5] Van der Maaten, *Learning a Parametric Embedding by Preserving Local Structure*, Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics, 2009.
- [6] Pedregosa, F. et al, *Manifold Learning*, recuperado de <https://scikit-learn.org/stable/modules/manifold.html> el 10 de Junio de 2020.