

Implementación de redes neuronales en manufactura aditiva.

Material de estudio "PLA"

Mtro. Diego Sánchez García
 Depto. Ingeniería
 Facultad de Estudios Superiores
 Cuautitlán
 CDMX, México
 dgosanchez@comunidad.unam.mx

Est. Angel Hernández Hernández
 Depto. Ingeniería
 Facultad de Estudios Superiores
 Cuautitlán
 CDMX, México
 hernandezhernandezangel694@gmail.com

Est. León Martínez Arantxa Quetzalli
 Depto. Ingeniería
 Facultad de Estudios Superiores
 Cuautitlán
 CDMX, México

Est. García Reyes José Armando
 Depto. Ingeniería
 Facultad de Estudios Superiores
 Cuautitlán
 CDMX, México

Abstract—In this paper we develop our passion for the study of materials together with the thriving and vast world of deep learning. We develop a neural network to predict the stress-strain curve of a 3D printed material by means of TensorFlow and JavaScript tools. The training data of the neural network are our own, generated in the laboratory by means of the tensile machine, the specimens from which these results were obtained were designed under standard, the 3D printing of each specimen was monitored and sectioned by means of various parameters considered. The results obtained were studied in depth to determine the reliability and efficiency of the implemented neural network along with the results of the same.

Keywords—neural network, stress-strain, standard, 3D print.

I. INTRODUCCIÓN

El interés por aplicar redes neuronales en el área de ciencias de materiales fue debido a lo costoso, tardado e inaccesible que puede llegar a ser realizar un ensayo de tracción para pruebas en manufactura aditiva, el adaptar redes neuronales nos evita diseñar una muestra, imprimir la muestra y finalmente someter la muestra a un ensayo de tracción. Evidentemente para el desarrollo de este proyecto ha sido necesario el estudio profundo del verdadero funcionamiento y operación de una red neuronal y de un riguroso método de ejecución en el ensayo de tracción.

El ensayo de tracción es una prueba fundamental que se realiza en un material para entender su comportamiento cuando se somete a fuerzas externas. Durante este ensayo se le aplica una fuerza que intenta estirar la probeta. A medida que se estira el material se registra la fuerza aplicada en cada momento y la cantidad que se estira la probeta.

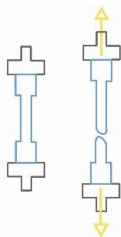


Ilustración 1 "Representación de una probeta sometida a un ensayo de tracción"

La grafica de tensión-deformación resultante de este ensayo proporciona la información que deseamos predecir con el desarrollo de este proyecto, la información resultante nos proporciona las propiedades mecánicas del material, los conceptos a comprender son:

Tensión. Donde la tensión es definida como la fuerza aplicada a partir de la superficie inicial de la probeta.

$$\sigma = \frac{F}{S_0} \dots (1)$$

Deformación. Que se define como el alargamiento partido por la longitud inicial de la probeta.

$$\varepsilon = \frac{\Delta L}{L_0} \dots (2)$$

Curva de tensión-deformación. Es un gráfico como el de la ilustración 2 con 2 regiones diferentes.

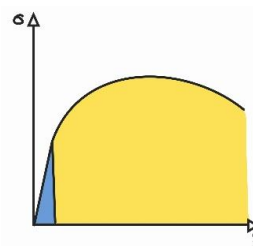


Ilustración 2 "Curva de tensión-deformación"

- La zona azul es la zona elástica, la cual es representada de una manera lineal. En esta zona elástica hay un rango de tensiones para el que se cumpla una correlación cuasi lineal entre tensión y deformación. Es llamado módulo de Young y en esta región el material se comporta como un resorte, es decir que recupera su forma inicial cuando se elimina la carga aplicada.
- La segunda zona amarilla, es la zona plástica, se aprecia una zona curva. Esta zona corresponde a la zona delimitada por tensiones superiores al límite elástico. No existe proporcionalidad entre las tensiones aplicadas y alargamientos producidos.

Lo que nos demuestra la gráfica es que hay una proporción directa entre la tensión aplicada a la deformación producida en el material. El ensayo nos ayuda a entender como un material reacciona a las fuerzas aplicadas y como se deforma bajo estas fuerzas. Esto es crucial para el diseño y selección de materiales en la ingeniería.

Los datos obtenidos en nuestras pruebas de tracción se pueden utilizar para comparar entre diferentes materiales o sobre el mismo material con un tipo (forma) de relleno, densidad de relleno y orientación de relleno diferentes, esto nos ayuda a verificar como el material va a resistir los esfuerzos a los que estará sometido cuando se utilice en alguna aplicación específica.

II. PRELIMINARES Y TRABAJOS RELACIONADOS

A. Diseño

El diseño de todas las probetas fue el mismo, utilizamos el software Fusion360 para el CAD, bajo las medidas y diseño como se presenta en la ilustración 3.

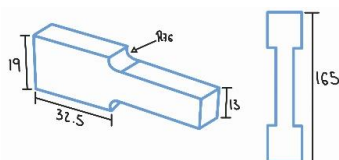


Ilustración 3 "Diseño de la probeta"

B. Manufactura aditiva

En esta parte preliminar del proyecto pasamos el diseño CAD hacia el código G para la compilación del diseño y su proceso de impresión, modificando los parámetros de impresión por los que nos hemos propuesto a observar en este proyecto, el software que ocupamos para compilar el diseño de impresión es UltiMaker Cura.

La impresión de todos los ejemplares se fue catalogando por parámetros como:

- Tipo de material
- Resolución
- Tipo de relleno
- Porcentaje de relleno

Todos los ejemplares fueron impresos con el mismo Diseño y justamente al momento de pasarlo al código G modificamos la impresión desde el software.

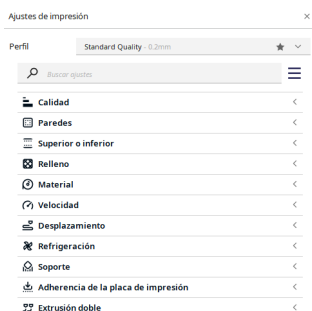


Ilustración 4 "Menú de configuración de impresión del software UltiMakerCura"

C. Ensayo de tracción

El ensayo de tracción se realizó sobre las probetas impresas, los ejemplares a los que se les considero aptos para el proyecto los catalogamos como:

- SLXXX.
- STXXX.

Donde el significado de:

- S = La resolución de la pieza **Standard Quality**
- L = Patrón de relleno **Lineal**
- T = Patrón de relleno **Triangular**
- XXX = Valor del porcentaje de relleno (%)

Por ejemplo.

- SL100 = Standard Lineal 100 %

El ensayo se elaboró como lo previamente explicado en la introducción de en este reporte. Se sujetó la probeta entre las mordazas y ejecutamos el ensayo de tracción como se puede apreciar en la ilustración 5.



Ilustración 5 "Ensayo de tracción"

III. EJECUCIÓN

A. Análisis de datos

El primer análisis de datos de realizó con la intención de conocer a profundidad el comportamiento de los datos generados por el ensayo de tracción.

Realizamos la exploración de los datos con el lenguaje de programación "Python", con el entorno de programación "Anaconda", una distribución libre utilizada en la ciencia de datos y aprendizaje automático que nos ayuda a procesar de una manera más sencilla y visual el manejo de grandes volúmenes de datos.

Las librerías utilizadas para este análisis de datos fueron:

- Pandas
- Matplotlib
- Seaborn
- Numpy
- Scipy
- Sklearn

Utilizando pandas, cargamos un archivo llamado "data.csv", este archivo contiene únicamente los datos de un ensayo, el ensayo nos arroja datos del Tiempo, Carga, Elongación, Disp, Tensión y Deformación.

Utilizamos un método de pandas que nos demuestre el tamaño de la matriz a analizar:

```
df_data.shape
(7019, 6)
Resultado
```

Determinamos la media de la tensión:

```
df_data['Stress'].mean()
29.022940589826284
Resultado
```

Sacamos un resumen de todo el set de datos:

```
df_data.describe()
Resultado
```

	Time	Load	Elong	Disp	Stress	Strain
count	7019.000000	7019.000000	7019.000000	7019.000000	7019.000000	7019.000000
mean	350.534125	0.685814	5.916663	-5.917015	29.022941	11.833366
std	202.433649	0.165099	3.417546	3.417542	6.586810	6.833082
min	0.000000	0.000000	0.000000	-11.833000	0.000000	0.000000
25%	175.257500	0.337000	2.958000	-8.760000	31.180000	5.914000
50%	350.534000	0.742000	5.916000	-5.917000	31.480000	11.832000
75%	525.810500	0.742000	8.876000	-2.958000	31.440000	17.752000
max	701.089000	0.832000	11.834000	-0.001000	35.125000	23.668000

Generamos relaciones entre todos los datos:

```
cormat = df_data.corr()
Resultado
```

	Time	Load	Elong	Disp	Stress	Strain
Time	1.000000	0.511460	1.000000	-1.000000	0.511456	1.000000
Load	0.511460	1.000000	0.511459	-0.511460	1.000000	0.511459
Elong	1.000000	0.511459	1.000000	-1.000000	0.511454	1.000000
Disp	-1.000000	-0.511460	-1.000000	1.000000	-0.511455	-1.000000
Stress	0.511456	1.000000	0.511454	-0.511455	1.000000	0.511454
Strain	1.000000	0.511459	1.000000	-1.000000	0.511454	1.000000

Demstrar esta relación entre las etiquetas de los datos nos permite darnos a la idea sobre el peso que tienen los datos respecto a otras etiquetas. A partir de la relación de datos generamos un mapa de calor para una demostración visual como lo demuestra la ilustración 6.

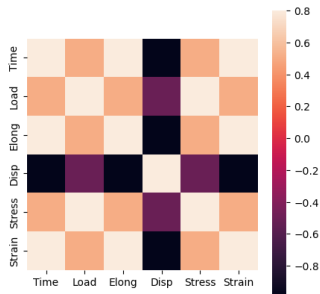


Ilustración 6 "Heatmap con el arreglo original"

El heatmap es un método de seaborn que nos demuestra a partir del set de datos original una relación tal cual y el set de datos este acomodado.

Para poder ver una relación respecto a una etiqueta en especial, ocupamos la librería numpy y pandas para generar de la matriz original una matriz transpuesta ordenada con los valores más grandes en columnas en orden descendente el arreglo se vería como en la ilustración 7.

```
array([[ 1.         ,  1.         ,  0.51145564,  0.5114544 ,  0.5114544 ,
        -0.51145537],
       [ 1.         ,  1.         ,  0.51146034,  0.5114591 ,  0.5114591 ,
        -0.51146007],
       [ 0.51145564,  0.51146034,  1.         ,  0.99999998,  0.99999998,
        -1.         ],
       [ 0.5114544 ,  0.5114591 ,  0.99999998,  1.         ,  1.         ,
        -0.99999998],
       [ 0.5114544 ,  0.5114591 ,  0.99999998,  1.         ,  1.         ,
        -0.99999998],
       [-0.51145537, -0.51146007, -1.         , -0.99999998, -0.99999998,
        1.         ]])
```

Ilustración 7 "Arreglo ordenado"

Generando una gráfica del arreglo ordenado se observaría como la ilustración 8.

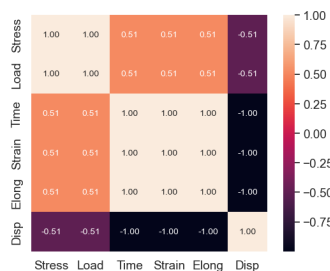


Ilustración 8 "Heatmap con el arreglo ordenado"

Lo que podemos apreciar en esta nueva grafica (sobre el 'Stress'), es una relación perfecta entre la tensión y la tensión lo cual es evidente dado que, si relacionamos los datos, cada valor del esfuerzo, tiene exactamente el mismo valor, o sea hay una relación 1:1. ¿Qué pasa con los demás datos para la tensión?

Bueno, la tensión respecto a:

- **Carga (Load):** Nos demuestra una relación unitaria.
- **Tiempo (Time):** Nos demuestra una relación media.
- **Deformación (Strain):** Nos demuestra una relación media.
- **Elongación (Elong):** Nos demuestra una relación media.
- **Disp:** No nos demuestra ningún tipo de relación

Para graficar todas las relaciones de las variables, ocupamos de seaborn un método que nos grafica la relación entre los valores como se aprecia en la ilustración 9.

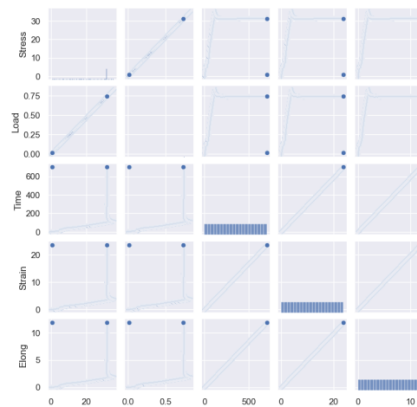


Ilustración 9 "Grafica pairplot"

B. Perceptrón

Ya con los datos analizados, podemos empezar a generar la idea de cómo y con qué datos implementar una red neuronal, dado que el objetivo de este estudio es generar una red neuronal que prediga la gráfica de tensión - deformación por medio de parámetros de impresión debemos recordar lo que es un perceptrón.

El perceptrón es una **única** neurona artificial creada por Frank Rosenblatt, la arquitectura propuesta se ve como la ilustración 10.

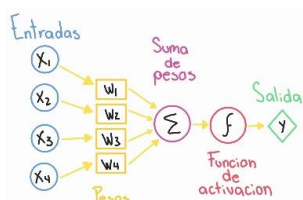


Ilustración 10 "Arquitectura el perceptrón"

Donde:

- X. Son las entradas donde básicamente son los valores numéricos que vamos a estar alimentando.
- W. Son los pesos, es donde las entradas se ponderan, cada entrada tiene su propio peso, parámetros que se multiplican.
- Σ. Es la suma de los pesos.
- f. Es la función de activación.
- Y. Salida.

Adaptando nuestro modelo a partir de la arquitectura del perceptrón de Frank Rosenblatt, demostrado en la ilustración 11.

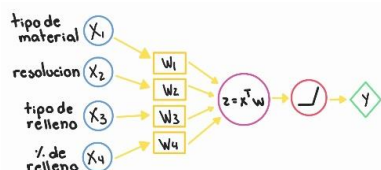


Ilustración 11 "Perceptrón propio"

Donde:

- Nuestras entradas son los parámetros de impresión. Recordando:
 - X_1 = Tipo de material
 - X_2 = Resolución
 - X_3 = Tipo de relleno
 - X_4 = Porcentaje de relleno
- Cada entrada tiene un cierto peso, un cierto nivel de importancia, así que multiplicaremos las entradas con un respectivo W .
- Entrando a la neurona, sumamos el resultado de todas las W .

$$z = x^T W$$

$$z = X_1 W_1 \dots X_4 W_4 + b$$

Donde b es el "bias", que son los parámetros que se suman.

- La función de activación es la parte de la neurona en la que convierte las funciones lineales generadas por la sumatoria de los pesos y las transforma en funciones no lineales al pasar por la función de activación. La función de activación que ocupamos en el desarrollo de nuestra red neuronal es una función ReLU.
- Y. Salida.

C. Red Neuronal

Dejando en claro que el perceptrón es una única neurona, al momento de juntar varios perceptrones por capas creamos nuestro perceptrón de multicapa también llamado red neuronal, teóricamente sabemos que una red neuronal se ve como la ilustración 12.

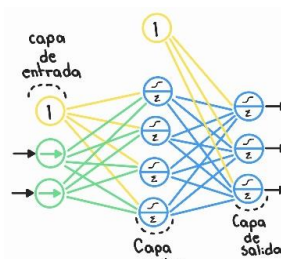


Ilustración 12 "Estructura de una red neuronal"

D. Keras y Tensor Flow

Para crear una red neuronal con estas librerías existen varios métodos, pero de manera resumida podemos contemplar los pasos más importantes que determinan a una red neuronal:

- Seleccionar la estructura de la red neuronal. Esto implica decidir el número de capas ocultas necesarias, cuantas neuronas habrá en cada capa, que tipo de función de activación se usará, etc.
- Definir los parámetros de la red. Esto puede incluir la tasa de aprendizaje, cantidad de epochs, batch size, función de optimización, etc.
- Utilizar TensorFlow para construir un modelo de la red neuronal. Esta herramienta nos proporciona una serie de funciones y clases que nos permiten definir y entrenar modelos de redes neuronales.
- Entrenamiento del modelo usando los datos como input para obtener una precisión óptima. Esto implica alimentar los datos de entrenamiento al modelo y ajustar los pesos y sesgos de la red a través del proceso de retro propagación y optimización del gradiente. En nuestro caso, el entrenamiento del modelo fueron las matrices de las 6 probetas con 3 diferentes porcentajes de relleno y 2 diferentes tipos de relleno, todo impreso con una calidad estándar.

IV. RESULTADOS

A. Graficas:

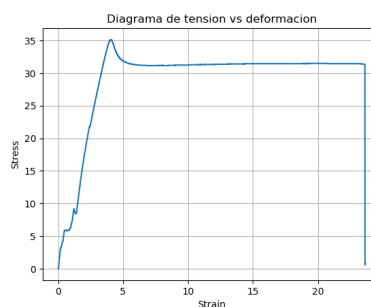


Ilustración 13 "Para una probeta SL100"

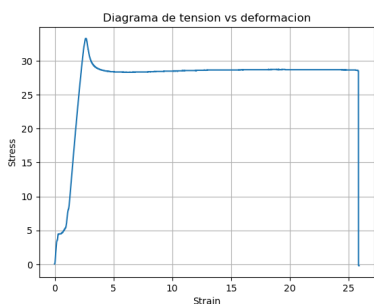


Ilustración 14 "Para una probeta ST100"

B. Conclusiones

Hemos considerado que mientras más variables físicas de impresión adjudiquemos a la composición de alguna pieza impresa que impliquen el comportamiento de esta, obtendremos una predicción y la formación de la curva aún más precisa, apegada a la real.

También, hemos considerado por medio del análisis de los datos que arroja la prueba de tracción que puede existir algún "overfitting" en los datos, por lo que aplicar una red

neuronal convolucional para el tratamiento directo de la grafica generada por el ensayo de tracción sería un poco mas efectiva al momento de generar la gráfica a predecir.

V. REFERENCIAS

- [1] Ivorra-Martinez, J., Quiles-Carrillo, L., Lascano, D., Ferrandiz, S., & Boronat, T. (2020). Effect of infill parameters on mechanical properties in additive manufacturing. *DYNA - Ingeniería e Industria*, 95(4), 412–417. <https://doi-org.pbidi.unam.mx:2443/10.6036/9674>
- [2] Monzón, M., Ortega, Z., Martínez, A., & Ortega, F. (2015). Standardization in additive manufacturing: activities carried out by international organizations and projects. *International Journal of Advanced Manufacturing Technology*, 76(5–8), 1111–1121. <https://doi-org.pbidi.unam.mx:2443/10.1007/s00170-014-6334-1>
- [3] Gómez-Alonso, J.-L., Allue, A., De-Marco, I., Retolaza, J., & Diez, G. (2022). Influencia del tipo de software de laminación, en las características de piezas impresas mediante tecnología de extrusión de filamento, con termoplástico reforzado con fibra corta. (Spanish). *DYNA - Ingeniería e Industria*, 97(3), 295–300. <https://doi-org.pbidi.unam.mx:2443/10.6036/10285>
- [4] Despeisse, M., & Ford, S. (2015). *The role of additive manufacturing in improving resource efficiency and sustainability* (Vol. 460). Springer New York LLC. https://doi-org.pbidi.unam.mx:2443/10.1007/978-3-319-22759-7_15