

Clasificación de haces de luz para el uso de energía por medio de redes neuronales

Alexi Vasquez Rodriguez, Leticia Flores Pulido,
Perfecto Malaquias Quintero, Carlos Perez Corona,
Francisco Javier Albores Velazco

Universidad Autónoma de Tlaxcala,
Facultad de Ciencias Básicas, Ingeniería y Tecnología,
México

{20181214, leticia.flores.p, perfectomalaquias.quintero.f,
carlos.perezc, franciscojavier.albores.v}@uatx.mx

Resumen. A pesar de los avances científicos en el área de la óptica aún existen fenómenos físicos de los cuales no se han investigado. En este caso de las redes neuronales en esta área no han sido del todo explotadas, los modelos de redes neuronales pueden facilitar la investigación de estas fuentes de luz. El propósito de este proyecto es crear una red neuronal, que, con base a características numéricas de las diversas fuentes de luz, se pueda clasificar eficazmente. La técnica utilizada para esta clasificación es Backpropagation. Esta es una técnica de propagación de errores hacia atrás, está programada en Python y en cada iteración se ingresarán patrones de entrada que serán los valores a clasificar. Se identificó que los resultados obtenidos se basan en la configuración de la red neuronal, esto se depende en gran medida de la forma en la que se conforme una arquitectura de red neuronal a la medida del problema.

Palabras clave: Redes neuronales, python, fuentes de luz, clasificación, inteligencia artificial.

Classification of Light Beams for Energy Use Using Neural Networks

Abstract. Despite scientific progress in optics area, there are still physical phenomena that have not been investigated. Since neural networks in this area have not been fully used, neural network models can make easier the research of these light sources. The purpose of this project is to create a neural network, which, based on numerical characteristics of the different light sources, can be effectively classified. The technique used for this classification is backpropagation. This is a backward error propagation technique programmed in Python where each iteration input patterns will be entered to be the values to classify. It was identified that the results obtained are based on the configuration

of the neural network, this depends on the way in which a neural network architecture tailored to the problem is formed.

Keywords: Neural networks, python, light sources, classification, artificial intelligence.

1. Introducción

Los haces de luz están compuestos por diferentes características, una de ellas es su longitud de onda, esta es una de las características en la que nos basamos para esta clasificación, esta onda contiene un campo eléctrico y un campo magnético, existen diferentes emisores de haces de luz, algunos de ellos son visibles y se puede determinar el color por la longitud de onda. Las redes neuronales son capaces de identificar y extraer las características que son realmente relevantes y deben utilizarse para resolver un problema, algo que escapa de las posibilidades de otras técnicas de aprendizaje automático [1].

Las redes neuronales nos ayudan a la ejecución de tareas complejas y en general específicas, nos ayudan a la clasificación de estos haces de luz ya que cuando se manda a una fuente de luz para su identificación este puede verse afectado por la luminosidad natural y es posible que arroje resultados erróneos. Estas redes neuronales están entrenadas con diferentes tipos de fuentes, este aprendizaje permite la reducción de operaciones que debería hacer para la identificación de las fuentes de luz.

2. Trabajos relacionados

En el trabajo realizado por Ramy [10], se presentó un proyecto en el que se clasifican diferentes fuentes de luz con eficiencia cuántica unitaria como fuentes de energía constante y aquellas con una eficiencia cuántica baja. Para la clasificación de una fuente de luz, la luz debe de crearse por conversión de un tipo de energía diferente. En el trabajo realizado por Ma et al. [9], se observó las características de las condiciones de luz, las condiciones de luz en interiores se describen habitualmente mediante niveles de iluminancia.

Estos mismos niveles de iluminancia, pueden ser generados por diferentes tipos de fuentes de luz, que irradian diferentes componentes. Este método fue evaluado para una serie de fuentes de luz comunes en interiores y bajo diferentes condiciones. En el artículo de Hloušek [4], se demostró que la precisión del DRNF (Detectores de resolución del número de fotones) informado mediante la medición de estadísticas de fotones para muchos estados diferentes de la luz, de los cuales se muestran 25 estados, que cubren varios números de fotones y g^2 valores. La obtención de fotones requiere mediciones repetidas utilizando un detector de resolución de número de fotones, algunos de los parámetros importantes para este son el rango, la velocidad y la precisión.

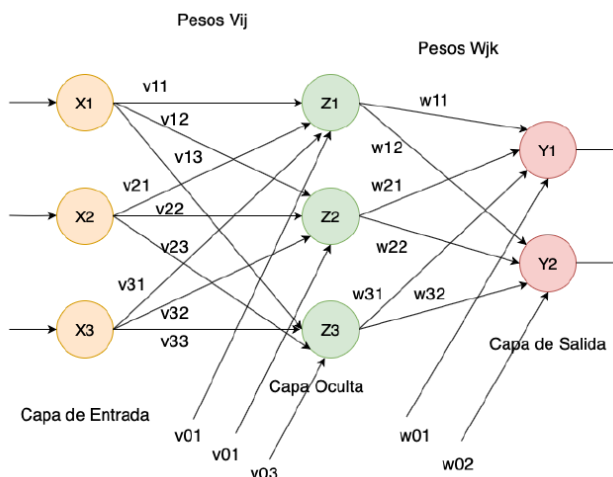


Fig. 1. Arquitectura básica de la red neuronal perceptrón multicapa. (Figura tomada de [7,8]).

En el trabajo hecho por Cohen [6], se mostró uno de los métodos de detección particulares, a saber, la teledetección láser, la detección y el alcance de la luz, está en continuo desarrollo, se han utilizado detectores monofónicos y detectores de resolución numérica de fotones para este fin. En el trabajo hecho por Kudyshev [5], se observó que el nano-ensamblaje determinista puede permitir la integración de dispositivos fotónicos cuánticos únicos en el chip. En este trabajo se implementó una clasificación supervisada basada en el aprendizaje automático de los emisores cuánticos como “simples” o “no simples” basándonos en sus datos de autocorrelación dispersos. En el trabajo realizado por Villegas [11], desarrollaron una mezcla de partículas, las cuales llevan mucho tiempo desempeñando un papel fundamental en muchas ramas de la ciencia y la tecnología, asistida por aprendizaje automático. Implementando varias redes neuronales conectadas en serie, cada una de las cuales realiza una predicción específica de las características del campo inicial.

3. Métodos y materiales

3.1. Algoritmo Backpropagation

La técnica Backpropagation, definido así por regresión de errores hacia atrás, es una técnica de redes neuronales que calcula el error de cada neurona de la última capa y lo propaga hacia atrás para saber el porcentaje de errores en toda la red. En la Figura 1, se muestra la arquitectura básica de la red neuronal perceptrón multicapa.

Esta arquitectura contiene una capa de entrada, una capa oculta, y una capa de salida, cada una de ellas interconectadas por flechas que representan los pesos

de las conexiones entre estas neuronas. Para los cálculos de esta red neuronal se necesitan las siguientes fórmulas tomadas [7], [8]. Para el cálculo de la capa oculta se necesita la siguiente ecuación (1), con su función de activación (2):

$$Z_j = v0_j + \sum x_i * v_{ij}, \quad (1)$$

$$Z_j \text{activada} = \frac{1}{1 + \exp(-Z_j)}. \quad (2)$$

Para el cálculo de la capa de salida se necesita la siguiente ecuación (3), con su función de activación (4):

$$Y_k = w0_k + \sum z_j * w_{jk}, \quad (3)$$

$$Y_k \text{activada} = \frac{1}{1 + \exp(-Y_k)}, \quad (4)$$

donde en la expresión (1), $v0_j$ y v_{ij} son los pesos de la capa de entrada a las capas ocultas, x es la entrada de los datos, y Z_j es la el conjunto de las neuronas ocultas. En la expresión (2), Z_j activada nos indicará si cada neurona oculta aprendió o no cada patrón de entrada. Por otro lado, en la expresión (3), tenemos que Y_k es la salida de clasificación, $w0_k$ son los pesos bias de la capa oculta a la capa de salida, z_j son las neuronas ocultas, y w_{jk} son los pesos de la capa oculta a la capa de salida.

En la expresión (4), Y_k activada nos indica si la neurona de salida aprendió o no el patrón de entrenamiento. En la expresión (5), se calcula el valor del error de salida (y). Cálculo del error de salida (y) e incremento de los pesos (w). Para calcular el error de salida se necesita la siguiente ecuación (5), donde t_k es la salida deseada del patrón 1, en este caso $t_1 = 1$:

$$\delta_k = (t_k - Y_k \text{activada}) * (Y_k \text{activada}) * [1 - Y_k \text{activada}]. \quad (5)$$

Para calcular los incremento de los pesos w que inciden en y necesitamos la siguiente ecuación (6), donde se hace el uso de un parámetro de aprendizaje alfa:

$$\Delta w_{jk} = \alpha * \delta_k * z_j \text{act}. \quad (6)$$

Para el cálculo del incremento de los bias se usa la siguiente ecuación (7):

$$\Delta w0_j = \alpha * \delta_j. \quad (7)$$

Cálculo del error oculto, activación e incremento de los pesos y bias v . Para este cálculo se sumará en cada unidad los errores ocultos delta que se obtuvieron en el paso anterior, usando la siguiente expresión:

$$\delta(z_j) = \sum \delta_j * w_{jk}. \quad (8)$$

Para la activación del error oculto se usa la siguiente expresión:

$$\delta(z_{j \text{act}}) = \delta z_j * z_{j \text{act}} * (1 - z_{j \text{act}}). \quad (9)$$

Para el cálculo del incremento de los peso v , se usa la siguiente ecuación:

$$\Delta v_{ij} = \alpha * \delta_j * x_i. \quad (10)$$

Para el cálculo del incremento de los bias se usa la siguiente ecuación:

$$\Delta v_{ij} = \alpha * \delta_j. \quad (11)$$

Finalmente para la actualización de los pesos w , v , bias w y bias v se usa la siguiente fórmula para todos los casos:

$$w_{ij} \text{ nuevo} = w_{ij} + \Delta w_{ij}. \quad (12)$$

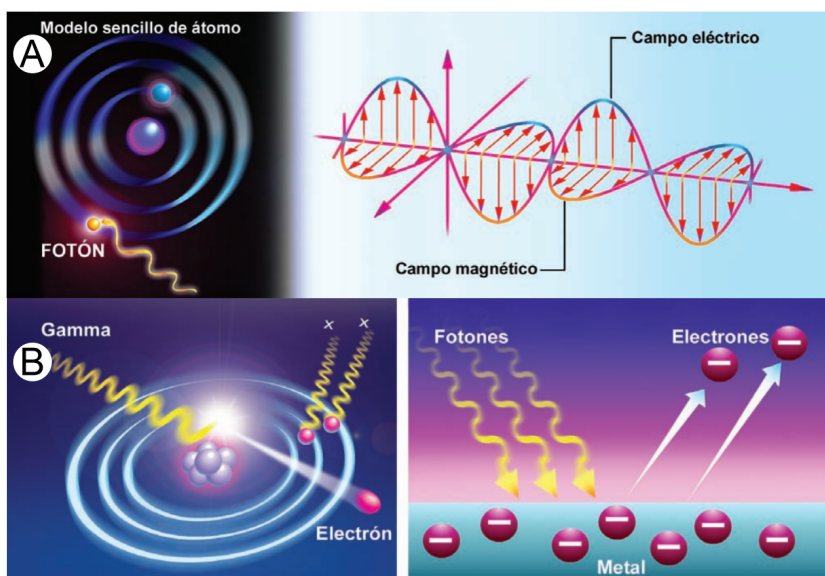


Fig. 2. Naturaleza de la luz y efecto fotoeléctrico. (Figura tomada de Bravo [2]).

3.2. Haces de luz

Los haces de luz conocidos por partículas u ondas de luz, en el libro de Hecht [3], habla sobre la definición de que la luz es un flujo constante de partículas submicroscópicas llamadas fotones. La luz es el resultado de un proceso cuántico que ocurre cuando un electrón se excita y pasa a un estado de energía alta y después la pierde. En la Figura 2 tomada de Bravo [2], la naturaleza de la luz.

- a) Se observa que la luz consiste en ondas electromagnéticas que se diferencian solo en su frecuencia. Esta onda electromagnética está compuesta

por dos perturbaciones: la primera es un campo eléctrico que vibra perpendicularmente a un campo magnético, también podemos observar el efecto fotoeléctrico.

- b) Este fenómeno consiste en que los electrones de la superficie de algunos metales se desprenden cuando se hacen incidir la luz sobre ellos.

4. Implementación

En la Figura 3 se observan los módulos que se desarrollarán las siguientes partes: Haces de luz: identificar diferentes haces de luz generados por fuentes naturales o artificiales tomando los datos de sus características de cada una de ellas. Técnica Backpropagation: se implementó la técnica Backpropagation, el objetivo es ajustar los pesos de una red neuronal en base a la tasa de error obtenida anteriormente, este ajuste de los errores garantiza una tasa de error más baja.

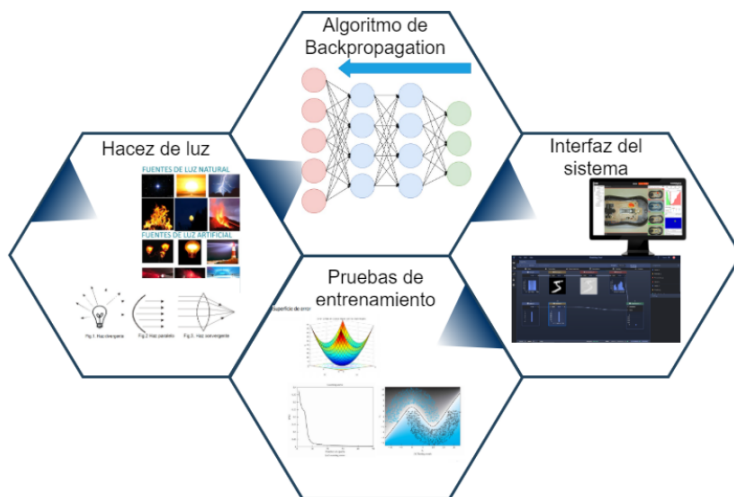


Fig. 3. Diagrama general del sistema de clasificación de haces de luz por medio de redes neuronales.

Prueba de entrenamiento: se analizan los resultados del entrenamiento en la cual se basa la técnica de Backpropagation. Interfaz del sistema: se desarrolló una interfaz para el usuario final, observaremos los resultados arrojados y el porcentaje de clasificación de estos haces de luz como se observa en la Figura 4.

En la Figura 4 se muestra la interfaz implementada para este proyecto. Se muestra la interfaz inicial, esta contiene principalmente un menú de opciones, en el menú de “Archivo” y “Ayuda”, en el apartado “Archivo” veremos 3 opciones

las cuales son: Abrir archivos entradas, Abrir archivo salidas y Salir, en la parte de “Ayuda” solo tendremos una opción de “Información” general del programa.

Dentro de la ventana tenemos los elementos con los que funciona la red neuronal, 1) tenemos la arquitectura propuesta, 2) ventana para visualizar los datos que ingresamos, 3) un botón de gráfica si se desea graficar los datos de entrada, 4) tenemos dos cuadros de entrada, uno se usará para ingresar el valor de Alpha y el siguiente para el número de las épocas y tenemos el botón de Backpropagation que nos ayudará a iniciar el entrenamiento de la red neuronal, 5) tenemos la gráfica de los resultados de salida, en la gráfica podemos dar clic en ella para generar otra ventana y verla más detalladamente, 6) se muestra la matriz de confusión de igual manera podemos dar clic para verlo más a detalle, 7) observamos algunas etiquetas las cuales nos muestran el porcentaje de la clasificación realizada y debajo de ellas un botón para observar gráficamente el cambio de los pesos que se hicieron durante el proceso.

5. Análisis de resultados

En la siguiente Tabla 1 se muestran los mejores resultados de esta clasificación en cada una de sus pruebas, se usaron 3 diferentes tipos de arquitecturas, la primera contiene 3 neuronas en la capa oculta, la segunda contiene solo 2 neuronas ocultas, y por último una arquitectura con 4 neuronas en la capa oculta.



Fig. 4. Diagrama general del sistema de Clasificación de haces de luz por medio de redes neuronales.

Cada una de estas arquitecturas se probaron con 3 diferentes valores para Alpha y el número de épocas, la primera configuración consistió en Alpha = 0.3 con un número de épocas = 10, la segunda configuración con un Alpha = 0.6 y

un número de épocas = 100, y la tercera configuración con un Alpha = 0.9 y un número de épocas = 1000.

En la Figura 5 se muestran los mejores resultados obtenidos por estas tres arquitecturas, cada una de ellas con diferente configuración, la arquitectura 1 se configuró con el parámetro de Alpha de 0.3 y 10 épocas, la arquitectura 2 con un Alpha de 0.6 y 100 épocas y finalmente la arquitectura 3 con un Alpha de 0.3 y 10 épocas. Se muestran los resultados con matrices de confusión para la mejor observación de estas clasificaciones. La parte A), corresponde a la arquitectura 1, la parte B), corresponde a la arquitectura 2, y la parte C), corresponde a la arquitectura 3.

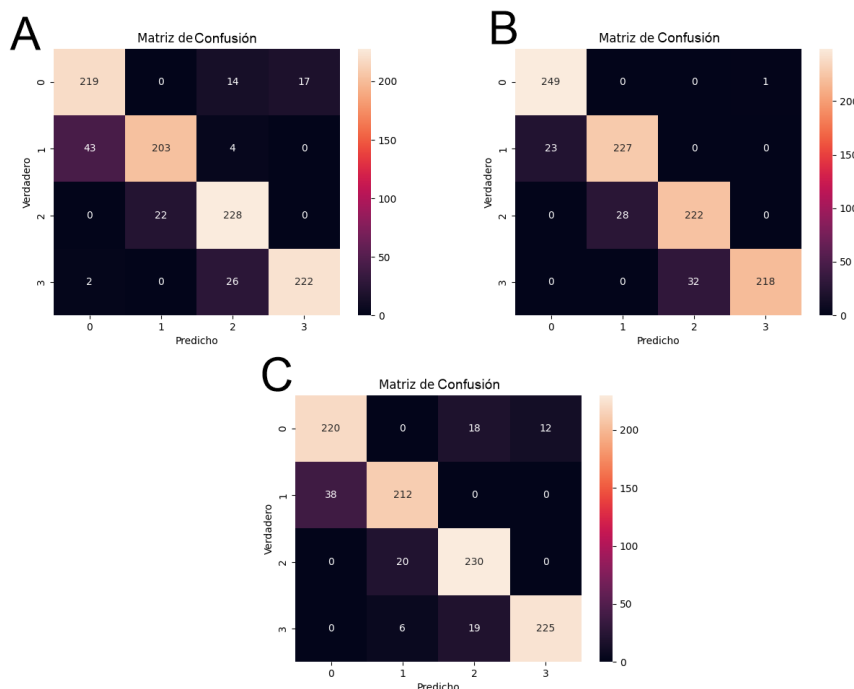


Fig. 5. Matrices de resultados para la mejor configuración del modelo utilizado.

6. Conclusiones

En este proyecto se hizo la clasificación de 4 diferentes tipos de haces de luz, cada uno de ellos con características numéricas en diferentes rangos. Las características que se tomaron fueron de longitud de onda, temperatura y potencia. Algunas de estas características se asemejan entre las fuentes de luz, como fue la longitud de onda. Una de las complicaciones al clasificar estas fuentes

de luz fue que algunas características tienen valores semejantes entre las fuentes y esto provoca una mala clasificación del modelo de red neuronal.

Por otra parte, la técnica Backpropagation es una técnica robusta y eficaz, usando algunos valores en los parámetros del modelo, usando una buena configuración de los parámetros podemos llegar a una perfecta clasificación de estos haces de luz. Usando este modelo con programación en Python, y además, creando el algoritmo desde cero podemos entender a mayores rasgos, que las redes neuronales se pueden implementar en problemas específicos, pero no todas las redes neuronales se adaptan a diferentes problemas.

Tabla 1. Tabla de resultados de todas las pruebas hechas.

Resultado de Pruebas					
Pruebas	Y1	Y2	Y3	Y4	Promedio
Prueba 1	87.6 %	81.2 %	91.2 %	88.8 %	87.2 %
Prueba 2	98.4 %	15.6 %	89.2 %	96.8 %	75 %
Prueba 3	100 %	0.0 %	0.0 %	100 %	50 %
Prueba 4	100 %	86.4 %	84.0 %	81.6 %	88 %
Prueba 5	99.6 %	90.8 %	88.8 %	87.2 %	91.6 %
Prueba 6	99.2 %	92.4 %	90.8 %	89.6 %	93 %
Prueba 7	88.0 %	84.8 %	92.0 %	90.0 %	88.7 %
Prueba 8	90.0 %	80.8 %	95.6 %	47.6 %	78.5 %
Prueba 9	100 %	0.0 %	0.0 %	100 %	50 %

Así se pudo observar que las topologías que mejor arrojaban el resultado eran las que tenían menos neuronas ocultas, esto podría deberse a que se cae en el sobre entrenamiento al agregar muchas neuronas en la capa oculta, si agregamos muchos nodos ocultos el valor de eficiencia del modelo se reduce bastante como se mostró en las pruebas.

Referencias

1. Berzal, F.: Redes neuronales & deep learning, Entrenamiento de redes neuronales artificiales. Spanish Edition (2018)
2. Bravo, A., de la Torre Bravo, A., Córdova, D.: Procedimientos endoscópicos en gastroenterología (2009)
3. Hecht, E.: Optica. Addison Wesley Longman (Pearson Education) (2000)
4. Hloušek, J., Dudka, M., Straka, I., Ježek, M.: Accurate detection of arbitrary photon statistics. Phys. Rev. Lett. 123, 153604 (2019)
5. Kudyshev, Z. A., Bogdanov, S. I., Isacsson, T., Kildishev, A. V., Boltasseva, A., Shalaev, V.M.: Rapid classification of quantum sources enabled by machine learning. Adv. Quantum Technol, vol. 3 (2020)

6. Cohen, L., Matekole, E .S., Sher, Y., Istrati, D., Eisenberg, H .S., Dowling, J.P.: Thresholded quantum LIDAR : Exploiting photon-number-resolving detection. In: OSA Quantum 2.0 Conference, OSA Technical Digest, Optical Society of America (2019)
7. Flores, L.: Antología de redes neuronales. Universidad Autónoma de Tlaxcala, México (2019)
8. Fausett, L. V.: Fundamentals of neural networks: Architectures, algorithms, and applications. Prentice-Hall (1994)
9. Ma, X., Bader, S., Oelmann, B.: Characterization of indoor light conditions by light source classification. *IEEE Sensors Journal*, vol. 17, no. 12, pp. 3884–3891 (2017)
10. Ramy G. S., El-Dardiry, S. F., Legendijk, A.: Classification of light sources and their interaction with active and passive environments (2011)
11. Villegas, A., Quiroz-Juarez, M., URen, A., Torres, J., León-Montiel, R.: Identification of particle mixtures using machine-learning-assisted laser diffraction analysis (2021)