



UNIVERSIDAD DE SONORA
"El Saber de mis Hijos hará mi Grandeza"

Red Neuronal para detectar incendios forestales



Curso de matemáticas para la maestría de Ciencia de Datos

Impartido por Jesus Fransisco Espinoza Fierro

Pablo Trinidad Chávez Amavizca

7 de junio de 2023

Clasificación de imágenes

En este tipo de redes neuronales se busca identificar y clasificar imágenes, ya sea reconocer rostros, categorizar prendas de vestir, diferenciar animales, entre muchas otras aplicaciones posibles.

En este caso, vamos a explorar los aspectos matemáticos involucrados en una red neuronal clasificadora de imágenes.

Un clasificador de imágenes

Nuestra red neuronal debe aprender a analizar imágenes, para lo cual se le proporcionan datos de entrada a cada neurona. En el caso de una imagen, se asignaría un píxel a cada neurona. Por ejemplo, si tenemos una imagen de 255 píxeles por 255 píxeles, se necesitan 65,025 neuronas en la red neuronal. Sin embargo, para simplificar el proceso, podemos reducir la imagen a su tamaño mínimo, es decir, 28 píxeles por 28 píxeles, lo que nos daría 784 píxeles entrando a 784 neuronas.

En cuanto a las neuronas de salida, esto dependerá de lo que estemos clasificando. Por ejemplo, si clasificamos animales, necesitaríamos la cantidad de neuronas correspondiente a la cantidad de animales que estamos clasificando. Podríamos tener 2 neuronas si solo clasificamos perros y gatos, o si clasificamos marcas, tendríamos tantas neuronas como marcas estamos clasificando.

La red neuronal que utilizaremos será un tipo de red neuronal convolucional, ya que es la que mejor se adapta a la clasificación de imágenes.

Funcionamiento matemático

Las redes neuronales convolucionales funcionan de la siguiente manera:

La función de una red neuronal convolucional (CNN, por sus siglas en inglés) es procesar y analizar datos de manera eficiente, especialmente en aplicaciones relacionadas con imágenes y datos de tipo matriz. Las CNN se utilizan comúnmente en tareas de reconocimiento de patrones, detección de objetos, segmentación de imágenes, clasificación y otras tareas de visión por computadora.

A diferencia de las redes neuronales tradicionales, las CNN están especialmente diseñadas para trabajar con datos estructurados en forma de matrices, como imágenes. La principal característica distintiva de las CNN es la capa de convolución, que realiza operaciones de convolución sobre los datos de entrada para extraer características relevantes.

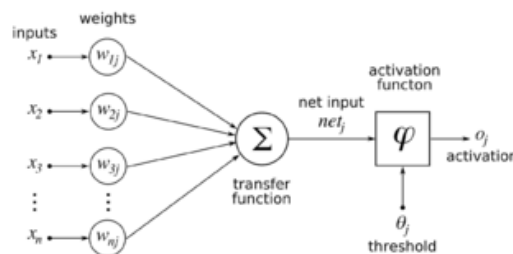
El proceso básico de una CNN implica la propagación hacia adelante (forward propagation) de los datos de entrada a través de varias capas, que generalmente incluyen capas de convolución, capas de agrupación (pooling), capas de activación y capas completamente conectadas. Durante la propagación hacia adelante, las capas de convolución aplican filtros a los datos de entrada para detectar características específicas, como bordes, texturas o formas. Las capas de agrupación reducen la dimensionalidad de las características extraídas, lo que ayuda a reducir la cantidad de parámetros y a extraer características invariantes a pequeñas variaciones espaciales.

Además, las capas de activación, como la función de activación ReLU (Rectified Linear Unit), se utilizan para introducir no linealidades en la red y permitir la representación de relaciones y patrones más complejos. Por último, las capas completamente conectadas agregan las características extraídas en un vector de características y realizan tareas de clasificación o regresión en función de los objetivos del modelo.

Durante el entrenamiento, las CNN ajustan los pesos y los sesgos de las diferentes capas para minimizar una función de pérdida, utilizando técnicas como el descenso de gradiente estocástico y la retropropagación del error.

$$z = \sum_{i=1}^n X_i W_i + b_i$$

$$a = F(z)$$



Como se explica en el texto anterior, cada neurona tiene asignados pesos para sus entradas, y cada entrada tiene un valor, además del sesgo conocido como bias. Para calcular las operaciones en una red neuronal de clasificación de imágenes, se sigue la siguiente fórmula:

...

salida = peso_1 * entrada_1 + peso_2 * entrada_2 + ... + peso_n * entrada_n + bias

...

Sin embargo, esta fórmula sólo produce resultados lineales. Para introducir no linealidad en la red neuronal, se utilizan funciones de activación.

Funciones de activación

Las funciones de activación se emplean para convertir los resultados en no lineales y evitar que se obtengan únicamente resultados lineales. Además, permiten mapear los valores de salida de la red neuronal a un rango determinado, como un sí o un no, o valores mapeados de -1 a 1 o de 0 a 1.

Existen dos tipos de funciones de activación:

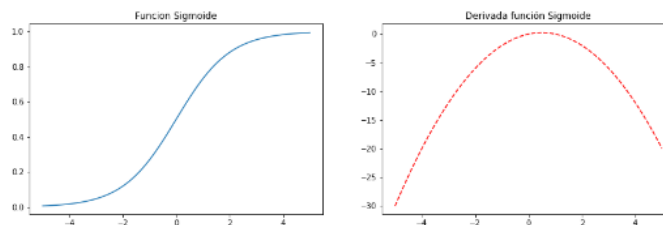
1. Funciones de activación lineales: Estas funciones ya no se utilizan en el Deep Learning, ya que la suma de varias funciones lineales sigue siendo una función lineal en sí misma.

2. Funciones de activación no lineales: Estas son las funciones que se emplean en las redes neuronales y permiten introducir no linealidad en los datos de salida. Algunos ejemplos de funciones de activación no lineales son:

- Función Sigmoide (Sigmoid): Convierte los valores en un rango entre 0 y 1.

$$\text{Sigmoide}(x) = \frac{1}{1+e^{-x}}$$

$$\text{sigmoide}'(x) = x(1 - x)$$



- Función ReLU (Rectified Linear Unit): Esta función toma el valor de entrada `x` y devuelve 0 si `x` es negativo, o el mismo valor de `x` si es positivo.

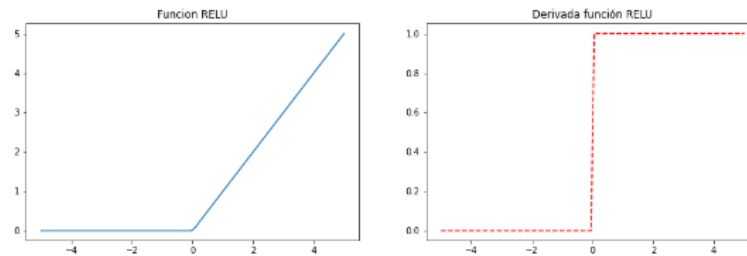
...

$$\text{ReLU}(x) = \max(0, x)$$

...

$$\text{relu}(x) = \max(0, x)$$

$$\text{relu}'(x) = 1. (x > 0)$$

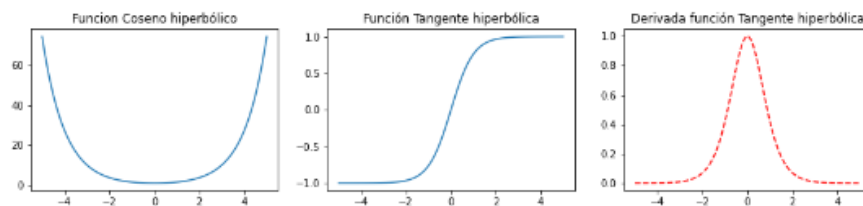


- Función Tanh (Tangente hiperbólica): Mapea los valores en un rango entre -1 y 1.

$$\cos(x) = \frac{e^{-x} + e^{-x}}{2}$$

$$\tan(x) = \frac{e^{-x} + e^{-x}}{e^{-x} + e^{-x}}$$

$$\tan'(x) = \frac{1}{\cos^2 x}$$



- Función Softmax: Utilizada en la capa de salida de una red neuronal para obtener probabilidades normalizadas entre 0 y 1 para cada clase en un problema de clasificación múltiple.

Estas funciones de activación permiten que la red neuronal capture patrones complejos y realice clasificaciones más precisas.

Correcto, una vez que obtenemos el resultado de una neurona, este resultado se pasa a través de la función de activación correspondiente. En este caso, mencionaste que utilizamos la función de activación ReLU.

Después de aplicar la función de activación a los resultados de las neuronas, llegamos al paso final del proceso conocido como PassForward (propagación hacia adelante). En este paso, evaluamos los resultados obtenidos (\hat{Y}_p) con los resultados esperados (Y_r), que forman parte de los datos de entrenamiento (x, y) .

La evaluación de los resultados obtenidos con los resultados esperados nos permite medir el rendimiento de la red neuronal y ajustar los pesos y sesgos de las neuronas durante el proceso de aprendizaje. Se utiliza una función de pérdida (loss function) para calcular la discrepancia entre los resultados obtenidos y los resultados esperados. A partir de esta discrepancia, se realiza el ajuste de los parámetros de la red neuronal mediante algoritmos de optimización, como la retropropagación del error (backpropagation).

Este proceso de evaluación y ajuste iterativo es fundamental para que la red neuronal aprenda a clasificar las imágenes de manera más precisa a medida que se expone a más datos de entrenamiento.

Para evaluar \hat{Y}_p y Y_r usamos una función de coste como:

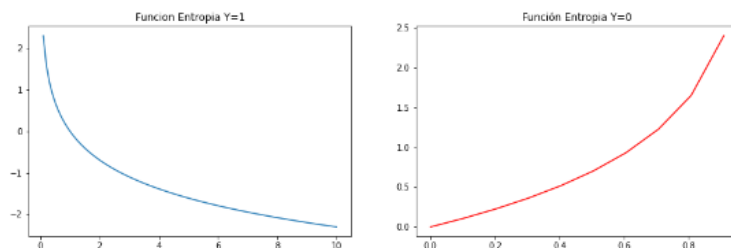
- MSE

$$MSE = \frac{1}{n} \sum_{l=l}^n (Y_i - \hat{Y}_i)^2$$

$$MSE = \sum_{i=l}^n (Y_i - \hat{Y}_i)$$

- Entropía cruzada binaria

$$H(y, P) = \{P = 1, -\log(P) \quad P = 0, -\log(1 - P)\}$$



Las funciones se pueden combinar

$$H(y, P) = \frac{1}{n} \sum_{l=l}^n (y \log(P) + (1 - y) \log(1 - P))$$

$$H(y, P) = - \left(\frac{y}{P} - \frac{1-y}{1-P} \right)$$

Exactamente, para evaluar la discrepancia entre los resultados obtenidos (\hat{Y}_p) y los resultados esperados (Y_r), se utiliza una función de costo (C), que puede representarse de la siguiente manera:

...

$$C = \text{coste}(\hat{Y}_p, Y_r)$$

...

Donde \hat{Y}_p es el valor predicho por nuestra red neuronal y Y_r es el valor real de los datos.

La elección de la función de costo depende del tipo de problema y de la naturaleza de los datos. Algunas funciones de costo comunes son:

- Mean Squared Error (MSE): Calcula el promedio de los errores al cuadrado entre los valores predichos y los valores reales. Se utiliza comúnmente en problemas de regresión.
- Cross-Entropy Loss: Utilizada en problemas de clasificación, especialmente cuando se emplea la función de activación Softmax en la capa de salida. Mide la divergencia entre la distribución de probabilidades predicha y la distribución real de las clases.

Una vez que se calcula el costo, el modelo ajusta sus parámetros, como las pesas y el sesgo (BIAS), para acercarse a la salida esperada Y . Este ajuste se realiza utilizando el algoritmo de retropropagación (backpropagation). El algoritmo de retropropagación calcula el gradiente de la función de costo con respecto a los parámetros del modelo y utiliza esta información para actualizar los parámetros de manera iterativa, buscando minimizar el costo y mejorar el rendimiento de la red neuronal en la tarea de clasificación de imágenes.

El algoritmo de retropropagación (backpropagation) se utiliza en conjunto con el algoritmo de descenso de gradiente (gradient descent) para ajustar las pesas (pesas) y sesgos (BIAS) de la red neuronal y minimizar la función de costo.

El objetivo del backpropagation es calcular los gradientes de la función de costo con respecto a los parámetros de la red (en este caso, los pesos y sesgos). Estos gradientes indican la sensibilidad al cambio del valor de salida de la red respecto a cambios en los valores de los parámetros, y nos proporcionan información sobre la dirección en la cual debemos ajustar los parámetros para minimizar el costo.

Para calcular los gradientes, se utiliza la regla de la cadena para derivadas. La derivada de la función de costo con respecto a un peso (w) se calcula multiplicando la derivada del costo con respecto a la suma ponderada (z) por la derivada de la suma ponderada con respecto al peso (w).

La derivada de la función de costo con respecto a un sesgo (BIAS) (b) se calcula multiplicando la derivada del costo con respecto a la suma ponderada (z) por la derivada de la suma ponderada con respecto al sesgo (BIAS) (b).

En resumen, las derivadas parciales de la función de costo con respecto a los pesos y sesgos nos proporcionan el gradiente local ($\partial C / \partial z$) en cada neurona. Este gradiente local se utiliza para computar el error en la capa interior y, finalmente, se actualizan los parámetros de peso utilizando el vector gradiente multiplicado por una tasa de aprendizaje (ϵ).

La tasa de aprendizaje (ϵ) determina el tamaño del paso que se toma en la dirección opuesta al gradiente para actualizar los parámetros. Es un hiperparámetro importante que influye en la convergencia y estabilidad del algoritmo de optimización.

En resumen, el backpropagation en combinación con el descenso de gradiente permite ajustar los pesos y sesgos de la red neuronal iterativamente, utilizando la información del gradiente para minimizar la función de costo y mejorar el rendimiento de la red en la tarea de clasificación de imágenes.

La problemática de los incendios forestales.

Los incendios forestales se distinguen por su gran amplitud y representan una problemática creciente que se ha visto agravada por el cambio climático. Se proyecta que el cambio climático y el cambio en el uso de la tierra aumenten la frecuencia y la intensidad de los incendios forestales en un 14% para 2030, un 30% para finales de 2050 e incluso un 50% para finales de siglo.

La relación entre el cambio climático y los incendios forestales se basa en el aumento de la sequía, las altas temperaturas del aire, la baja humedad relativa, los relámpagos y los fuertes vientos. Estos factores

contribuyen a temporadas de incendios más cálidas, secas y prolongadas. Los incendios forestales no solo afectan negativamente a los humanos, sino que también representan una amenaza para la vida silvestre y los hábitats naturales, pudiendo llevar a especies animales y vegetales al borde de la extinción.

Para abordar esta problemática, se propone un sistema de detección de incendios forestales mediante el uso de imágenes y una red neuronal convolucional. Este sistema utilizará datos obtenidos de Kaggle, los cuales son de uso libre. Se dividirán los datos en tres conjuntos: entrenamiento, prueba y validación. El conjunto de entrenamiento constará de 30,250 imágenes, mientras que los conjuntos de prueba y validación tendrán 6,300 imágenes cada uno.

La red neuronal convolucional que se utilizará en este sistema tendrá las siguientes características:

1. Tamaño de las imágenes de entrada: Las imágenes serán de 32x32 píxeles, lo que implica que cada imagen se representará como una matriz de 32 filas y 32 columnas.
2. Neuronas de entrada: La red contará con 1,024 neuronas de entrada. Cada imagen se convertirá en un vector unidimensional de longitud 1,024 antes de ingresar a la red neuronal, lo cual permite procesar la matriz de píxeles de la imagen.
3. Capas de convolución: Se utilizarán tres pares de capas de agrupación máxima (max pooling) con filtros de 32, 64 y 128 respectivamente. Cada capa de convolución aplicará operaciones de convolución para extraer características de las imágenes, generando mapas de características.
 - Primera capa de convolución: Esta primera capa cuenta con 32 filtros. Cada filtro recorre la imagen de entrada, realizando operaciones de convolución para extraer características. La salida de la capa será una matriz con 32 canales o mapas de características.
 - Primera capa de agrupación máxima(max pooling): Después de la capa de convolución, se aplica la capa de agrupación máxima con tamaño de núcleo de 2x2 píxeles. El trabajo de esta capa es reducir la dimensionalidad de cada mapa de características por medio de la selección del valor máximo de cada región de 2x2 píxeles. Su salida será una matriz con 32 mapas de características reducidas.
 - Segunda capa de convolución: En la segunda capa de convolución hay 64 filtros. Justo como en la primera capa, cada filtro se desliza por la salida de la capa anterior y se

realizan las operaciones para extraer las características. En esta capa la salida será una matriz con 64 mapas de características.

- Segunda capa de agrupación máxima(max pooling): Después de la segunda capa de convolución, se aplica una capa de agrupación máxima igual con un núcleo de 2x2 píxeles. Su salida será una matriz de 64 mapas de características reducidas.
 - Tercera capa de convolución: En esta capa tenemos 128 filtros. Se aplica la convolución a los mapas de características de la anterior capa y nos genera una matriz con 128 mapas de características.
 - Tercera capa de agrupación máxima(max pooling): Después de la tercera capa de convolución, se aplica la tercera capa de agrupación máxima, también con un núcleo de 2x2 píxeles. Esta capa hará que se reduzca la dimensionalidad nuevamente y entregará una matriz con 128 mapas de características reducidas.
4. Función de activación ReLU: Después de cada capa de convolución, se aplicará la función de activación ReLU, que mantendrá los valores positivos sin cambios y establecerá en cero los valores negativos. Esto introduce no linealidad en la red y permite la extracción de características más complejas.
 5. Capa de agrupación máxima (max pooling): Después de cada capa de convolución, se aplicará una capa de agrupación máxima con un tamaño de núcleo de 2x2 píxeles. Esta capa reducirá la dimensionalidad de los mapas de características seleccionando el valor máximo en cada región de 2x2 píxeles.
 6. Dropout: Después de cada capa de agrupación máxima o convolución, se aplicará un dropout con una probabilidad de retención del 60% ($1 - 0.4$). Esto implica que cada neurona tiene una probabilidad del 60% de mantenerse activa durante cada iteración de entrenamiento. El dropout ayuda a evitar el sobreajuste al desactivar aleatoriamente algunas neuronas, evitando así la dependencia excesiva entre ellas.
 - Las neuronas desactivadas se excluyen de manera aleatoria del cálculo de las salidas y de la propagación hacia las capas posteriores. Con la finalidad de evitar la dependencia excesiva entre las neuronas y promover la generalización, ya que la red no puede depender demasiado de ninguna neurona en particular.
 - Durante la inferencia o evaluación, no aplicamos el dropout. En su lugar se ajustan los pesos de las neuronas de acuerdo con la probabilidad de retención utilizada durante el entrenamiento. Así garantiza que las salidas se puedan escalar adecuadamente y se obtengan resultados más precisos.

7. Capa de salida: La capa de salida utilizará la función de activación softmax, que normaliza las salidas de las neuronas en probabilidades. Esto producirá una distribución de probabilidad para las diferentes clases de salida, lo que permite determinar la probabilidad de que una imagen contenga un incendio forestal.

En conclusión, el sistema propuesto utilizará una red neuronal convolucional con imágenes de 32x32 píxeles como entrada. La red consta de capas de convolución, capas de agrupación máxima, funciones de activación ReLU, dropout y una capa de salida con función de activación softmax. Este enfoque permitirá detectar la presencia de incendios forestales en áreas remotas o de difícil acceso, contribuyendo a su detección temprana y prevención.

Referencias:

- UN, Environment (2022) «El número de incendios forestales aumentará un 50% en 2100 y los gobiernos no están preparados, advierten los expertos». recuperado de [\[https://www.unep.org/es/noticias-y-reportajes/comunicado-de-prensa/el-numero-de-incendios-forestales-aumentara-un-50-en\]](https://www.unep.org/es/noticias-y-reportajes/comunicado-de-prensa/el-numero-de-incendios-forestales-aumentara-un-50-en)
- Calvo, J. (2020) «Crear redes neuronales desde las matemáticas». *European Valley Institution*. Recuperado de [\[https://www.europeanvalley.es/noticias/crear-red-neuronal-desde-las-matematicas/\]](https://www.europeanvalley.es/noticias/crear-red-neuronal-desde-las-matematicas/)
- Abdelghani, A. (2023) «Wildfire prediction dataset (Satellite Images)». Recuperado de [\[https://www.kaggle.com/datasets/abdelghaniaaba/wildfire-prediction-dataset?select=valid\]](https://www.kaggle.com/datasets/abdelghaniaaba/wildfire-prediction-dataset?select=valid)
- Mathworks «¿Qué es una red neuronal convolucional?». Recuperado de [\[https://es.mathworks.com/discovery/convolutional-neural-network-matlab.html\]](https://es.mathworks.com/discovery/convolutional-neural-network-matlab.html)