



AWAKELAB

BASECAMP

Ciencia de Datos

Módulo: Fundamentos de deep learning.

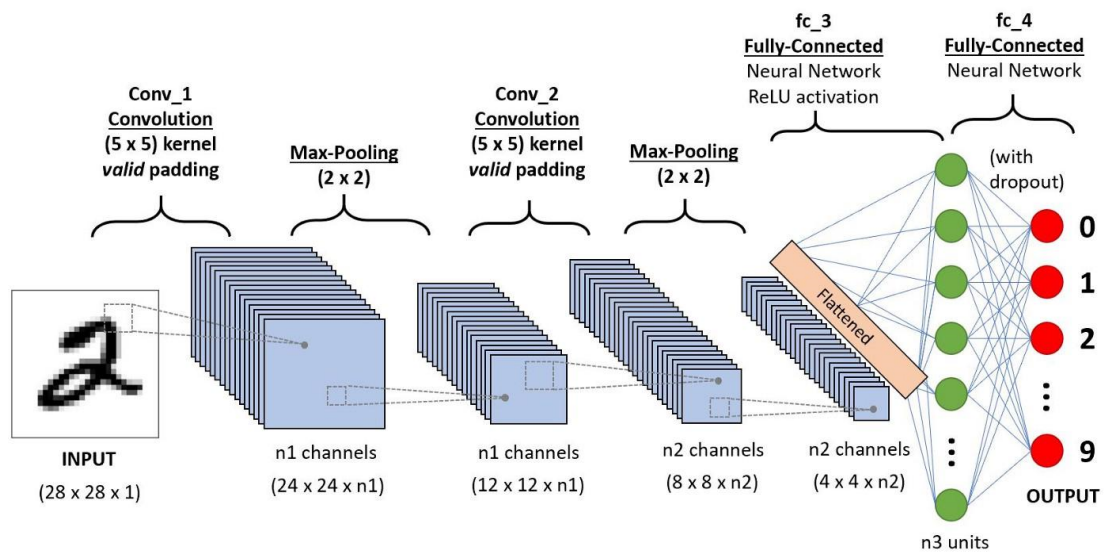
Aprendizaje Esperado

3. Implementar un modelo predictivo utilizando redes neuronales convolutivas para el reconocimiento de imágenes en Python

Redes Neuronales Convolucionales

La inteligencia artificial ha sido testigo de un crecimiento monumental en la reducción de la brecha entre las capacidades de los humanos y las máquinas. Tanto los investigadores como los entusiastas trabajan en numerosos aspectos del campo para hacer que sucedan cosas asombrosas. Una de muchas de estas áreas es el dominio de la visión artificial.

La agenda para este campo es permitir que las máquinas vean el mundo como lo hacen los humanos, lo perciban de manera similar e incluso usen el conocimiento para una multitud de tareas como reconocimiento de imágenes y videos, análisis y clasificación de imágenes, recreación de medios, sistemas de recomendación, procesamiento de lenguaje natural, etc. Los avances en visión artificial con aprendizaje profundo se han construido y perfeccionado con el tiempo, principalmente sobre un algoritmo en particular: una red neuronal convolucional.



Una red neuronal convolucional (ConvNet/CNN) es un algoritmo de aprendizaje profundo que puede tomar una imagen de entrada, asignar importancia (pesos y sesgos aprendibles) a varios aspectos/objetos en la imagen y ser capaz de diferenciar uno de otro. El procesamiento previo requerido en una ConvNet es mucho más bajo en comparación con otros algoritmos de clasificación. Mientras que en los métodos primitivos los filtros están diseñados a mano, con suficiente capacitación, ConvNets tiene la capacidad de aprender estos filtros/características.

La arquitectura de una ConvNet es análoga a la del patrón de conectividad de las neuronas en el cerebro humano y se inspiró en la organización de Visual Cortex. Las neuronas individuales responden a los estímulos solo en una región restringida del campo visual conocida como Campo Receptivo. Una colección de tales campos se superponen para cubrir toda el área visual.

¿Qué es una convolución?

Las capas convolucionales son los principales bloques de construcción utilizados en las redes neuronales convolucionales.

Una convolución es la simple aplicación de un filtro a una entrada que resulta en una activación. La aplicación repetida del mismo filtro a una

entrada da como resultado un mapa de activaciones llamado mapa de características, que indica las ubicaciones y la fuerza de una característica detectada en una entrada, como una imagen.

La innovación de las redes neuronales convolucionales es la capacidad de aprender automáticamente una gran cantidad de filtros en paralelo específicos para un conjunto de datos de entrenamiento bajo las restricciones de un problema de modelado predictivo específico, como la clasificación de imágenes. El resultado son características muy específicas que se pueden detectar en cualquier parte de las imágenes de entrada.

Campo de Utilización de RN

¿Cómo hacer uso de las redes neuronales convolucionales? Las empresas suelen estar en la búsqueda de una guía de redes neuronales convolucionales, que esté especialmente enfocada en las aplicaciones de las CNN para enriquecer la vida de las personas.

1. Decodificación de reconocimiento facial

El reconocimiento facial se descompone mediante una red neuronal convolucional en los siguientes componentes principales:



- Identificar cada rostro en la imagen.
- Centrarse en cada rostro a pesar de los factores externos, como la luz, el ángulo, la pose, etc.
- Identificación de características únicas
- Comparar todos los datos recopilados con datos ya existentes en la base de datos para hacer coincidir una cara con un nombre.

También se sigue un proceso similar para el etiquetado de escenas.

2. Análisis de documentos



Las redes neuronales convolucionales también se pueden utilizar para el análisis de documentos. Esto no solo es útil para el análisis de escritura a mano, sino que también tiene un gran interés en los reconocedores. Para que una máquina pueda escanear la escritura de una persona y luego compararla con la amplia base de datos que tiene, debe ejecutar casi un millón de comandos por minuto. Se dice que con el uso de CNN y modelos y algoritmos más nuevos, la tasa de error se ha reducido a un mínimo de 0,4% a nivel de carácter, aunque aún no se ha visto ampliamente su prueba completa.

3. Colecciones Históricas y Ambientales

Las CNN también se utilizan para fines más complejos, como las colecciones de historia natural. Estas colecciones actúan como actores clave en la documentación de partes importantes de la historia, como la biodiversidad, la evolución, la pérdida de hábitat, la invasión biológica y el cambio climático.



4. Comprender el clima

Las CNN se pueden utilizar para desempeñar un papel importante en la lucha contra el cambio climático, especialmente para comprender las razones por las que vemos cambios tan drásticos y cómo podemos experimentar para frenar el efecto. Se dice que los datos en tales colecciones de historia natural también pueden proporcionar una mayor comprensión social y científica, pero esto requeriría recursos humanos calificados, como investigadores que puedan visitar físicamente este tipo



de depósitos. Existe la necesidad de más mano de obra para llevar a cabo experimentos más profundos en este campo.

5. Áreas grises

La introducción del área gris en las CNN se plantea para proporcionar una imagen mucho más realista del mundo real. Actualmente, las CNN funcionan en gran medida exactamente como una máquina, viendo un valor verdadero y falso para cada pregunta. Sin embargo, como humanos, entendemos que el mundo real se desarrolla en mil tonos de gris. Permitir que la máquina comprenda y procese una lógica más difusa la ayudará a comprender el área gris en la que vivimos los humanos y en la que nos esforzamos por trabajar. Esto ayudará a las CNN a obtener una visión más holística de lo que ven los humanos.



6. Publicidad



Las CNN ya han aportado un mundo de diferencia a la publicidad con la introducción de la compra programática y la publicidad personalizada basada en datos.

7. Otros campos interesantes

Las CNN están preparadas para ser el futuro con su introducción en automóviles sin conductor, robots que pueden imitar el comportamiento humano, asistentes en proyectos de mapeo del genoma humano, predicción de terremotos y desastres naturales, y tal vez incluso autodiagnósticos de problemas médicos. Por lo tanto, ni siquiera tendría que conducir hasta una clínica o programar una cita con un médico para asegurarse de que su ataque de estornudos o fiebre alta sea solo una simple gripe y no síntomas de alguna enfermedad rara. Un problema en el que están trabajando los investigadores con las CNN es la detección del cáncer cerebral. La detección temprana del cáncer cerebral puede resultar un gran paso para salvar más vidas afectadas por esta enfermedad.



Etapas en las Redes Neuronales Convolutivas (CNN)

Las CNN están estructuradas de manera diferente en comparación con una red neuronal normal. En una red neuronal normal, cada capa consta de un conjunto de neuronas. Cada capa está conectada a todas las neuronas de la capa anterior. La forma en que funcionan las redes neuronales convolucionales es que tienen capas tridimensionales de ancho, alto y profundidad. Todas las neuronas de una capa en particular no están conectadas a las neuronas de la capa anterior. En cambio, una capa solo está conectada a una pequeña porción de neuronas en la capa anterior.

Comencemos con la capa superior:

- **La capa matemática**

La capa superior se percibe como la capa matemática. Es esencialmente la capa convolucional y se ocupa de comprender el patrón numérico que ve. Supongamos que la primera posición en esta capa comienza a aplicar un filtro alrededor de la esquina superior izquierda de la imagen. El filtro

también se conoce como neurona o núcleo. Lee esa parte de la imagen y forma una conclusión de una matriz de números, multiplica la matriz y deduce un solo número de este proceso.

- Este único número representa la esquina superior izquierda que la capa convolucional acaba de leer de la imagen. La parte de la imagen que escanea el filtro es el campo receptivo. El filtro luego se mueve a la derecha 1 unidad y comienza el mismo proceso nuevamente. De esta manera, la capa convolucional lee la imagen completa y asigna un solo número a cada unidad. Estos datos se almacenan en una matriz 3D. En esencia, todo este proceso funciona como el cerebro humano. A lo que nos referimos como campo receptivo en el mundo de las CNN es al campo visual en el mundo de la biología humana. El filtro actúa como la corteza visual que contiene pequeñas regiones de células que se dirigen a la lectura de áreas específicas del campo visual.

- **La capa de unidad lineal rectificada**

La siguiente capa encontrada es la Capa de unidad lineal rectificada (ReLU). Aquí es donde tienen lugar las funciones de activación. La función de activación se establece inicialmente en un umbral cero. El gradiente de activación solo funciona en 0 y 1 y no incluye gradientes intermedios como sus predecesores. Debido a su forma lineal, no saturada, se dice que las ReLU ayudan mucho en el gradiente decreciente de error. Sin embargo, debido a la naturaleza frágil de una ReLU, es posible que hasta el 40 % de su red esté inactiva en un conjunto de datos de entrenamiento.

- En un nivel superior, la primera capa en una red neuronal convolucional profunda es la capa convolucional, seguida de una unidad lineal rectificada, seguida de otra capa convolucional, y luego alternando unidades lineales rectificadas y capas de agrupación junto con solo una capa convolucional más. Si bien el proceso que sigue la primera

capa convolucional es bastante sencillo, el proceso se vuelve más complejo a medida que avanzamos por las capas, ya que las capas convolucionales ya no se ocupan de una imagen simple. Se trata de la salida luego procesada de las matemáticas iniciales aplicadas en cada nivel.

- **La capa completamente conectada**

Como ocurre con cualquier producto terminado, se requiere tener una capa final que abarque todas las complejidades interiores. Esta capa es la capa de finalización en una red neuronal convolucional. Toma la salida final de la capa anterior (ya sea una ReLU o una capa convolucional) y proporciona una salida vectorial N-dimensional. 'N' aquí significa el número de clases de las que elige el programa. Por ejemplo, si el programa está viendo imágenes de caballos, buscará características de alto nivel como las 4 patas, los cascos, la cola o el hocico. Esta capa totalmente conectada observará las características de alto nivel y las conectará con la imagen, dando así el resultado de una clasificación de un caballo.

Implementación de CNN en Python

Usaremos el conjunto de datos de clasificación Mnist Digit.

Los datos no estructurados pueden ser texto, imágenes, videos, audios, básicamente los datos que no están en un formato definido o estructurado. No hay filas, columnas, valores o características predefinidas en los datos no estructurados, y es más complicado que los datos estructurados. Los modelos de aprendizaje profundo están diseñados de tal manera que imitan la capacidad del cerebro humano y, por lo tanto, son más robustos para resolver este tipo de problemas.

Aquí, vamos a utilizar los datos de dígitos escritos a **mano del MNIST que contienen las imágenes de los dígitos escritos a mano (referencia [0])**.

```
#importar las bibliotecas necesarias

from tensorflow.keras.datasets import mnist
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D
from tensorflow.keras.layers import MaxPool2D
from tensorflow.keras.layers import Flatten
from tensorflow.keras.layers import Dropout
from tensorflow.keras.layers import Dense
```

```
#cargando data

(X_train,y_train) , (X_test,y_test)=mnist.load_data()

#remodelando data

X_train = X_train.reshape((X_train.shape[0],
X_train.shape[1], X_train.shape[2], 1))

X_test =
X_test.reshape((X_test.shape[0],X_test.shape[1],X_test.shape[2],1))

#comprobación de la forma después de la remodelación

print(X_train.shape)

print(X_test.shape)

#normalizar los valores de los píxeles
```

```
X_train=X_train/255

X_test=X_test/255

Downloading data from
https://storage.googleapis.com/tensorflow/tf-keras-
datasets/mnist.npz

11493376/11490434 [=====] -
0s 0us/step

11501568/11490434 [=====] -
0s 0us/step

(60000, 28, 28, 1)
(10000, 28, 28, 1)
```

```
#definiendo el modelo

model=Sequential()

#agregando capa convolucion

model.add(Conv2D(32,(3,3),activation='relu',input_shape=(28,28,1)))

#agregando capa de agrupación

model.add(MaxPool2D(2,2))

#agregando capa totalmente conectada

model.add(Flatten())

model.add(Dense(100,activation='relu'))

#agregar capa de salida
```

```
model.add(Dense(10,activation='softmax'))

#compilación del modelo

model.compile(loss='sparse_categorical_crossentropy',
optimizer='adam',metrics=['accuracy'])

#ajuste del modelo

model.fit(X_train,y_train,epochs=10)
```

Epoch 1/10

```
1875/1875 [=====] - 37s
19ms/step - loss: 0.1532 - accuracy: 0.9539
```

Epoch 2/10

```
1875/1875 [=====] - 35s
19ms/step - loss: 0.0520 - accuracy: 0.9844
```

Epoch 3/10

```
1875/1875 [=====] - 35s
19ms/step - loss: 0.0329 - accuracy: 0.9898
```

Epoch 4/10

```
1875/1875 [=====] - 35s
18ms/step - loss: 0.0227 - accuracy: 0.9925
```

Epoch 5/10

```
1875/1875 [=====] - 34s
18ms/step - loss: 0.0159 - accuracy: 0.9949
```

Epoch 6/10

```
1875/1875 [=====] - 34s
18ms/step - loss: 0.0114 - accuracy: 0.9962
```

Epoch 7/10

1875/1875 [=====] - 34s

18ms/step - loss: 0.0082 - accuracy: 0.9973

Epoch 8/10

1875/1875 [=====] - 35s

19ms/step - loss: 0.0068 - accuracy: 0.9976

Epoch 9/10

1875/1875 [=====] - 35s

19ms/step - loss: 0.0064 - accuracy: 0.9980

Epoch 10/10

1875/1875 [=====] - 34s

18ms/step - loss: 0.0044 - accuracy: 0.9984

<keras.callbacks.History at 0x7f1a84b60a90>

#evaluando el modelo

model.evaluate(X_test,y_test)

313/313 [=====] - 2s

7ms/step - loss: 0.0643 - accuracy: 0.9866

[0.06428956240415573, 0.9865999817848206]

Hemos cubierto algunos elementos importantes de CNN, mientras que aún quedan muchos, como el relleno, el aumento de datos, más detalles

sobre Stride, pero como el aprendizaje profundo es un tema profundo e interminable.

Referencias

[0] Implementación de práctica ANN

<https://www.analyticsvidhya.com/blog/2021/08/implementing-artificial-neural-network-on-unstructured-data/>

[1] Red neruonal convolucional

https://es.wikipedia.org/wiki/Red_neuronal_convolutiva

[2] Introducción a Redes neuronales convolucionales

<https://bootcampai.medium.com/redes-neuronales-convolucionales-5e0ce960caf8>

[3] Creacion de redes en Keras

<http://blog.hadsonpar.com/2021/08/crear-una-red-neuronal-convolutiva-en.html>

Material Complementario

[1] Redes neuronales convolutivas CNN

<https://www.youtube.com/watch?v=ns2L2T6wvAY>

[2] Redes neuronales usando librería Keras

<https://www.youtube.com/watch?v=L2k1GUyislo>