

Introduction to Deep Learning & Neural Networks with keras

Week 4

Shallow Versus Deep Neural Networks

1. Shallow Versus Deep Neural Networks

- Una red neuronal con una capa oculta se considera una red neural superficial. La entrada de este tipo de capas suele ser un vector.
- Una red con muchas capas ocultas y un gran número de neuronas en cada capa se considera una red neuronal profundo red neuronal. En este caso, las entradas pueden ser imágenes, texto o cualquier otro dato.

El auge repentino en el campo de deep learning se puede atribuir a tres factores principales:

1. Avance en el campo.
2. Otra razón principal es la disponibilidad de datos. Las redes neuronales profundas funcionan mejor cuando se entrenan con grandes cantidades de datos, ya que las redes neuronales aprenden de los datos de entrenamiento, entonces grandes cantidades de datos tienen que ser utilizados para evitar el overfitting de los datos de entrenamiento.
3. Finalmente, el poder computacional. Con las GPU super potentes de NVIDIA, ahora somos capaces de entrenar redes neuronales muy profundas en una cantidad tremenda de datos en cuestión de horas en lugar de días o semanas.

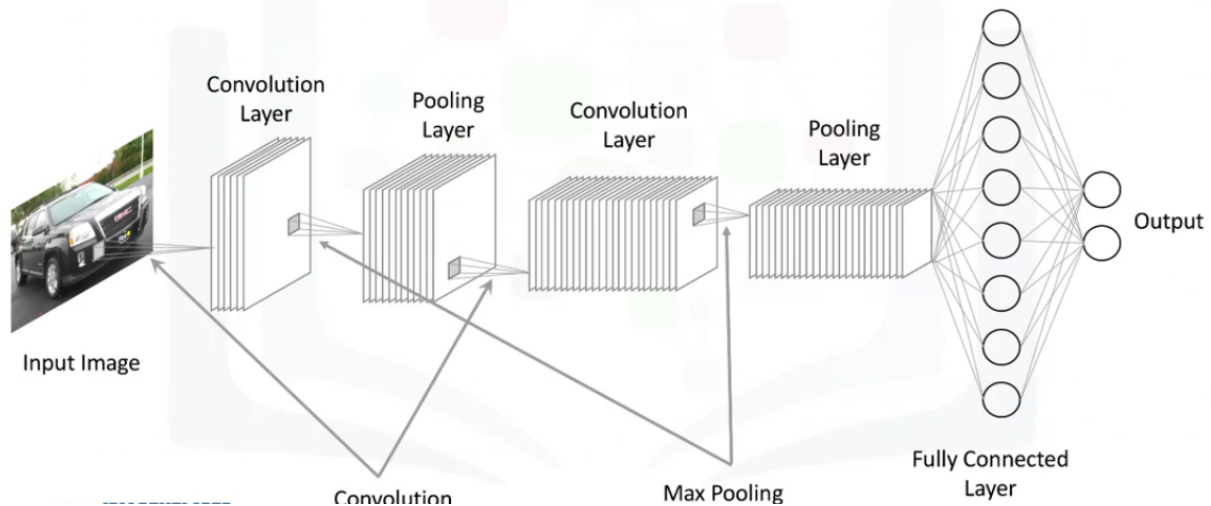
Supervised Learning Models

1. Convolutional Neural Networks

Las redes neuronales convolucionales están compuestas por neuronas, que necesitan tener los pesos y sesgos optimizados. Cada neurona combina las entradas que recibe mediante el cálculo del producto matricial entre cada entrada y el peso correspondiente antes de que se ajuste a la entrada total resultante en una función de activación, ReLU más probable.

CNN para abreviar, tienen como entradas imágenes, lo que nos permite incorporar ciertas propiedades en su arquitectura. Estas propiedades hacen que el paso de propagación hacia adelante sea mucho más eficiente y reduce enormemente la cantidad de parámetros en la red. Por lo tanto, las CNN son las mejores para resolver problemas relacionados con el reconocimiento de imagen, la detección de objetos y otras aplicaciones de visión por computadora.

En la imagen podemos ver una arquitectura típica de una red neuronal convolucional.



La red consta de una serie de capas convolucionales, ReLU y agrupaciones, así como un número de capas totalmente conectadas que son necesarias antes de que se genere la salida.

La entrada a una red neuronal convolucional, por otro lado, es mayormente un $(n \times m \times 1)$ para imágenes en escala de grises o un $(n \times m \times 3)$ para imágenes coloreadas

- En la **capa convolucional**, básicamente definimos filtros y calculamos la convolución entre los filtros definidos y cada una de las tres imágenes (RGB). Además consta de ReLU que filtran la salida del paso convolucional pasando sólo valores positivos y girando cualquier valor negativo a 0.
- La siguiente capa es la **capa de agrupación**: El objetivo principal de la capa de agrupación es reducir las dimensiones espaciales de los datos que se propagan a través de la red. Hay dos tipos de agrupación que son ampliamente utilizados en las redes neuronales convolucionales:
 - Agrupación máx: es el más común.
 - Agrupación promedio. En max-pooling
- Finalmente, **la capa de salida**: aplanamos la salida de la última capa convolucional y conectamos cada nodo de la capa actual con cada otro nodo de la siguiente capa. Esta capa básicamente toma como entrada la salida de la capa anterior, ya sea una capa convolucional, ReLU, o capa de agrupación, y genera un vector de n-dimensional, donde n es el número de clases pertenecientes al problema que se está tratando. Por ejemplo, si está creando una red para clasificar imágenes de dígitos, la dimensión n sería 10, ya que hay 10 dígitos.

2. Recurrent Neural Networks

Las redes neuronales recurrentes o (RNN) son redes con bucles que no sólo toman una nueva entrada a la vez, sino que también toman como entrada la salida del punto anterior que fue alimentado en la red.

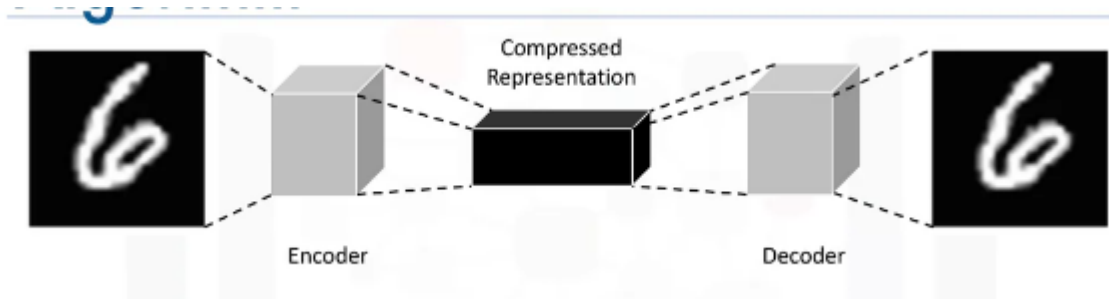
Como resultado, las redes neuronales recurrentes son muy buenas para modelar patrones y secuencias de datos, como textos, genomas, escritura a mano y mercados de valores. Estos algoritmos toman en cuenta el tiempo y la secuencia, lo que significa que tienen una dimensión temporal. Un tipo popular de red neuronal recurrente muy es el modelo de memoria a corto plazo o el modelo (LSTM) para abreviar. Se ha utilizado con éxito para muchas aplicaciones incluyendo generación de imágenes, donde un modelo entrenado en muchas imágenes se utiliza para generar nuevas imágenes novedosas.

Unsupervised Learning Models

1. Autoencoders

La codificación automática es un algoritmo de compresión de datos donde las funciones de compresión y descompresión se aprenden automáticamente de los datos, en lugar de ser diseñadas por un humano. Dichos autocodificadores se construyen utilizando redes neuronales. Los autocodificadores son específicos de datos, lo que significa que solo podrán comprimir datos similares a los que han sido entrenados. Por lo tanto, un autoencoder entrenado en imágenes de coches haría un pobre trabajo de comprimir imágenes de edificios, porque las características que aprendería serían específicas del vehículo o del coche.

Algunas aplicaciones interesantes de autocodificadores son la descodificación de datos y la reducción de dimensionalidad para la visualización de datos. Aquí está la arquitectura de un autocodificador.



Debido a las funciones de activación no lineales en redes neuronales, los autocodificadores pueden aprender proyecciones de datos que son más interesantes que un análisis de componentes principales PCA u otras técnicas básicas, que pueden manejar sólo transformaciones lineales.

Un tipo muy popular de autocodificadores es el Restricted Boltzmann Machines o (RBMs) para abreviar. Los RBMs se han utilizado correctamente para varias aplicaciones, incluida la corrección de conjuntos de datos desequilibrados. Debido a que RBMs aprenden la entrada para poder regenerarla, entonces pueden aprender la distribución de la clase minoritaria en un conjunto de datos de desequilibrio, y luego generar más puntos de datos de esa clase, transformando el conjunto de datos de desequilibrio en un conjunto de datos balanceados.

Otra aplicación popular de Restricted Boltzmann Machines es la extracción automática de características de datos especialmente no estructurados.