

Apuntes del 20 de Mayo de 2025

Brandon Adones Calderon Cubero
Escuela Ingenieria en Computacion)
Instituto Tecnologico de Costa Rica
Cartago, Costa Rica
brandonacubero@estudiantec.cr

I. RESPUESTAS DEL QUIZ

- 1) **Describa qué es una capa de convolución y una capa de pooling.**

Respuesta:

Una **capa de convolución** es aquella que extrae los *features* (características) relevantes de la imagen mediante la aplicación de *kernels* (filtros).

La **capa de pooling** es aquella que tiene como objetivo reducir la dimensionalidad de la imagen a procesar, manteniendo la dimensionalidad de la profundidad.

- 2) **Mencione dos ventajas por las cuales se prefiere aplicar pequeñas convoluciones anidadas en lugar de una convolución grande, si su salida fuera equivalente.**

Respuesta:

Dos ventajas son: (1) la menor cantidad de parámetros necesarios para extraer la información, y (2) una mayor capacidad de analizar la imagen, ya que se utilizan funciones no lineales entre cada capa.

- 3) **Describa parte por parte un módulo de Inception.**

- Convolución 1x1:** Aplica filtros de tamaño 1x1 sobre la entrada.
- Convolución 3x3:** Aplica filtros 3x3 para captar patrones espaciales medianos.
- Convolución 5x5:** Aplica filtros 5x5 para captar patrones más grandes.
- 3x3 MaxPooling:** Realiza una operación de pooling sobre la entrada.

- 4) **Describa qué es cada variable (m, k, p, s) en la siguiente fórmula para el cálculo de dimensiones.**

$$\frac{m - k + 2p}{s} + 1 = \text{Dimensión resultado}$$

Respuesta:

- m = tamaño de la entrada (input)
- k = tamaño del kernel (filtro)
- p = padding (relleno)
- s = stride (salto del filtro)

II. REPASO

A. Autoencoders

Los autoencoders hay debate si son de aprendizaje no supervisado o si son de aprendizaje supervisado, ya que estos se basan en la reconstrucción de una imagen.

El autoencoder es supervisado con la misma imagen que se le dio como entrada, ya que este reconstruye la data que venía de entrada, hay una nube gris en determinar si es supervisado o si no es supervisado ya que solo usa la entrada mas no hay etiquetas.

En la literatura en su mayoría se le asigna como no supervisado ya que usa la misma entrada de x para su aprendizaje.

Hay 3 partes que lo componen:

Encoder: extrae características de imagen

Espacio latente o cuello de botella: vector de características es distribuido

Decoder: A partir de los embeddings se reconstruye la imagen

Nota*: La imagen no queda exactamente igual, ya que esta es reconstruida a partir de una imagen a partir de un vector mas pequeño, mientras más grande sea el vector, más similar sera la imagen. Este concepto se puede aplicar a cualquier tipo de red neuronal.

1) *Reduccion de dimensionalidad*: La reducción de dimensionalidad permite representar la entrada de un vector mas compacto, ademas que es más poderosa en comparación con PCA y permite reconstruir la información con menos perdida.

2) *Deteccion de anomalias*: El modelo se entren para reconstrucción de datos e una tarea tomando en cuenta su estado positivo, que es usado para validar transferencias bancarias correctas y es de alta fidelidad. Ademas aprende la representación latente de casos positivos. Si una transferencia es suficientemente distinta a la normal, el auto-encoder (encoder) generara un vector latente deficiente, el cual no se puede reconstruir.

El function loss sera muy alto al momento de la reconstrucción y se debe definir un umbral para marcarlo como fraudulento.

3) *Procesamiento de imagenes*: Se puede utilizar para la compresión de imágenes, reducir el ruido de las mismas ademas de aumentar la resolución en la cuales a partir de imágenes de baja resolución se aumenta la resolución usando técnicas como upscaling y así mismo se puede hacer reconocimiento facial en imágenes de baja resolución.

III. ENCODER

Este es un conjunto de bloques convolucionales seguidos de módulos de pooling que es aplicar técnicas de convolución hasta reducir una imagen y tener feature maps que se convierten un vector. Este se encarga de la extracción de características de imagen y comprime la información. La idea es obtener un vector latente el cual se le conoce como downsampling, que es lo que le permite representar la imagen de entrada en algún punto en el espacio.

IV. CUELLO DE BOTELLA

Es la parte mas importante y pequeña del modelo, si este espacio latente esta mal hecho el autoencoder no va a funcionar correctamente.

Posee representación en espacio latente.

Restringe el flujo de la información proveniente del encoder al autoencoder, que es la reducción de la dimensionalidad pasado al decoder.

A medida que el entrenamiento mejora, se van separando mas los datos formando tipo clusters de las cosas mas similares juntas.

V. DECODER

Este vendría siendo las convoluciones que hacen outsampling y a partir de aquí se empieza a reconstruir la imagen basado en el vector latente.

VI. HIPERPARAMETROS A CONSIDERAR

El tamaño de la codificación (vector latente) va a depender de la arquitectura entre mas entradas tenga el vector, mayor es la información y es mayor la compresión (a mayor tamaño mayor costo de procesamiento).

El número de capaz a utilizar sigue el principio de a mayor número de capaz mejores resultados se van a obtener, pero es más complejo es el modelo y se puede overfitear y la capacidad de inferencia disminuye.

Para la reconstrucción es totalmente dependiente de la entrada y la salida, se usa MSE para comparar cada uno de los pixeles que deberían de ser iguales tanto en la entrada como en la salida y el binaryCrossEntropy entre los valores de rangos [0,1].

VII. TIPOS DE AUTOENCODER

A. Undercomplete Autoencoder

Este tipo de modelo toma la imagen de entrada e intenta predecir la misma imagen de salida.

La dimensión del espacio latente es menor que el espacio de entrada.

B. Sparse Autoencoder

No va a tratar de reducir la dimensionalidad, sino apagar ciertas neuronas para aprender a hacer el vector tipo haciendo dropout a cada una de las neuronas. Selectivamente apaga las neuronas de la red.

C. Denoising Autoencoder

Este es un modelo utilizado para remover el ruido de una imagen y su salida no tiene la imagen de entrada

D. Variational Autoencoder

Los autoencoders estándar aprenden a representar los datos en un espacio latente, pero este espacio no es necesariamente continuo ni estructurado.

Generalmente, el vector latente de un autoencoder tradicional no sigue ninguna distribución específica; simplemente contiene valores aprendidos durante el entrenamiento. Esto puede dificultar la **interpolación** (es decir, estimar valores intermedios entre puntos conocidos del espacio latente), ya que no existe una estructura probabilística que relacione los distintos puntos.

Para solucionar este problema, se utiliza el **Variational Autoencoder**. Este aprende a representar los datos en el espacio latente de manera probabilística, forzando a que los vectores latentes sigan una distribución. Esto permite interpolar y generar datos nuevos de forma más sencilla y coherente, haciendo el espacio latente continuo y útil para tareas generativas.

Para hacer el variational autoencoder es necesario producir que los vectores sigan una distribución normal con media μ y desviación estándar específica. El codificador produce 2 vectores para hacer la distribución la cual es la representación de μ y σ estándar. Se hace una reparametrización, donde se toman las muestras de μ y σ estándar para poder producir el vector latente y una vez hecho esto se toma el vector latente y se alimenta el decoder. Nota* la función de perdida es la suma del Binary Cross Entropy y la de Kullback-leibler Divergence.

VIII. AUTOENCODER SEGMENTACION

Se usa la arquitectura U-net porque tiene forma de U y se basa en un autoencoder.

IX. AUTOENCODER SEGMENTACIÓN

Se utiliza la arquitectura **U-Net** porque tiene forma de "U" y se basa en el principio de los autoencoders. U-Net es especialmente popular en tareas de segmentación de imágenes médicas y científicas.

La arquitectura U-Net consta de dos partes principales:

- **Encoder:** Similar a un autoencoder tradicional, esta parte se encarga de extraer las características principales de la imagen de entrada, reduciendo progresivamente su tamaño a través de capas de convolución y pooling.
- **Decoder:** Esta parte reconstruye la imagen segmentada a partir de la representación comprimida, utilizando capas de upsampling que permite hacer la segmentacion y la entrada y salida son diferentes porque la salida se compara con el **ground truth** de la segmentacion aplicada.

Una característica clave de U-Net es el uso de skip connections, que unen las capas del encoder y decoder en niveles correspondientes. Esto permite reconstruir mucho mejor las imágenes que se están procesando.