Zero shot Learning: Clasificación de imágenes

William Guadalupe
FC-UNI
Lima, Perú
wguadalupeq@uni.pe

Victor Chavez
FC-UNI
Lima, Perú
vchavezb@uni.pe

Stefano Olivieri
FC-UNI
Lima, Perú
solivierir@uni.pe

***Resumen*—Durante los ultimas decadas se observa que para poder entrenar los algoritmos de machine learning se necesita de un gran conjunto de datos de entrenamiento,sin embargo no siempre se puede obtener un gran conjunto de datos y debidamente etiquetado ,Zero-Shot learning generalmente funcionan asociando clases observadas anteriormente y no observadas .**

En el presente proyecto se llegara a mostrar el algoritmo Zero-Shot Learning y los metodos que usa como lo son Embedding-based y Generative-base, la cual va a ser entrenada con un dataset de animales con sus respectivas a caracteristicas y/o descripciones en donde nuestra entrada es tanto las imagenes de los animales y asi como sus caracteristicas y al añadir una nueva imagen se va a predecir los atributos y una imagen no entrenada de un animal

Índice de Términos–Zero-shot Learning,Pytorch,Resnet18

I. Introducción

I-A. Problematica

Durante las últimas décadas, las máquinas se han vuelto mucho más inteligentes, pero sin un conjunto de datos de entrenamiento debidamente etiquetado de las clases vistas, no pueden distinguir entre dos objetos similares. Sin embargo, los humanos son capaces de identificar aproximadamente 30.000 categorías de objetos básicos. En el aprendizaje automático, esto se considera el problema de Zero-shot Learning (ZSL).

I-B. ¿Que es Zero-Shot Learning?

El Zero-Shot Learning es poder resolver un problema a pesar de no haber recibido ningún ejemplo sobre este. Veamos un ejemplo concreto, imagina reconocer una categoría de objeto en fotos sin haber visto nunca una foto de ese tipo de objeto antes. Ya sea que usted ha leído una descripción muy detallada de un animal como el gato, es posible que pueda saber qué es un gato en una iamgen la primera vez que lo vea debido a lo que anteriormente ha conocio sobre este animal

I-C. ¿Por que Zero-Shot Learning?

El desarrollo de modelos de aprendizaje automático pueden realizar funciones predictivas en datos que antes no se habían visto, esto se ha convertido en un área de investigación importante conocida como Zero-Shot Learning. Las personas tenemos la tendencia a ser buenos para reconocer cosas a nuestro alrededor que no hemos vimos antes y Zero-Shot Learning nos ofrece una posible manera para imitar esta capacidad humana.

I-D. Objetivos

I-D1. Objetivos Generales:

- Aprender el desarrollo del algoritmo Zero Shot Learning.

I-D2. Objetivos Especificos:

- Determinar un dataset adecuado para el algoritmo Zero Shot Learning.
- Implementar el código para la extracción de caracteristicas.
- Implementa el código de la funcion de proyeccion del espacion visual al espacio semántico.
- Validar si el modelo tiene una precisión aceptable.

I-E. Herramientas y Métodos

I-E1. Herramientas: Para el desarrollo del proyecto utilizamos el lenguaje de programación Python en cual usamos las librerías que serán detallados a continuación:

- Numpy: Es una biblioteca de Python que proporciona un objeto de matriz multidimensional, varios objetos derivados (como matrices y matrices enmascaradas) y una variedad de rutinas para operaciones rápidas en matrices, que incluyen manipulación matemática, lógica, de formas, clasificación, selección, E / S. [9]
- PyTorch : Es una biblioteca de aprendizaje automático de código abierto basada en la biblioteca de Torch, utilizado para aplicaciones que implementan cosas como visión artificial y procesamiento de lenguajes naturales. Las librerias usadas fueron torch.nn , torchvision, torch.utils.
- Resnet 18 Es una red neuronal convolucional que tiene 18 capas de profundidad. Puede cargar una versión previamente entrenada de la red entrenada en más de un millón de imágenes desde la base de datos ImageNet. La red previamente entrenada puede clasificar imágenes en 1000 categorías de objetos, como teclado, mouse, bolígrafo y muchos animales.
- Nitron Es un servicio en la web, la cual nos permite cargar nuestros modelos en un formato onxx para

visualizar la composicion interna del modelo a usar, esto ayuda a entender de manera gráfica la relacion de entradas-salidas en la red.

I-F. Hipótesis

¿Es posible determinar el tipo de animal a partir de una imagen que no es entrenado, basandonos en imagenes entrenadas que comparten caracteristicas semanticas en comun?

I-G. Justificación

La justificación por la cual debemos usar este algoritmo es que nos ahorra el tiempo de entrenamiento de una gran cantidad de clases no conocidas, ya que solo prescindible conocer los atributo del conjunto de datos y por relaciones semanticas podemos inferir el resultado de la predicción.

II. Estado del Arte

II-A. An embarrassingly simple approach to zero-shot learning ,Bernardino Romera-Paredes,Philip H. S. Torr

[1]

La clasificación automática es el primer problema considerado en el aprendizaje automático, por lo que se ha estudiado y analizado, dando lugar a una amplia variedad de enfoques de clasificación que han demostrado su utilidad en muchas áreas como la visión computacional y la clasificación de documentos.

Sin embargo, estos enfoques generalmente no pueden abordar escenarios desafiantes en los que pueden aparecer nuevas clases en la etapa de aprendizaje.

II-B. A Review of Generalized Zero-Shot Learning Methods

[2] En este trabajo con los avances recientes en el procesamiento de imágenes y la visión computacional, los modelos de aprendizaje profundo (Deep Learning) han alcanzado una gran popularidad debido a su capacidad para proporcionar una solución integral desde la extracción de características hasta la clasificación. A pesar de su éxito, los modelos Deep Learning tradicionales requieren entrenamiento en una gran cantidad de datos etiquetados para cada clase, junto con una gran cantidad de muestras. En este sentido, es un desafío recolectar muestras etiquetadas a gran escala. Como ejemplo, ImageNet, que es un gran conjunto de datos, contiene 14 millones de imágenes con 21,814 clases en las que muchas clases contienen solo unas pocas imágenes. Además, los modelos Deep Learning estándar únicamente pueden reconocer muestras pertenecientes a las clases que se han visto durante la fase de entrenamiento y no pueden manejar muestras de clases no antes vistas.

Si bien en muchos escenarios del mundo real, es posible que no haya una cantidad significativa de muestras etiquetadas para todas las clases. Por un lado, la anotación detallada de una gran cantidad de muestras es laboriosa y requiere un conocimiento experto del dominio. Por otro lado, muchas categorías carecen de suficiente data en este escenario nos encontramos en muchas situaciones. Esto sucede cuando se trata de un conjunto creciente de clases, como la detección de nuevas especies de animales, Muestras etiquetadas, por ejemplo, aves en peligro de extinción, u observadas en progreso, por ejemplo, COVID-19, o no cubiertas durante el entrenamiento, pero aparecen en la fase de prueba.

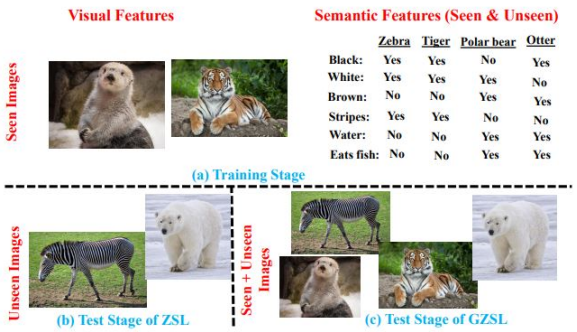


Figura 1. Zero-Shot

[2]

II-C. Train Once, Test Anywhere: Zero-Shot Learning For Text Classification

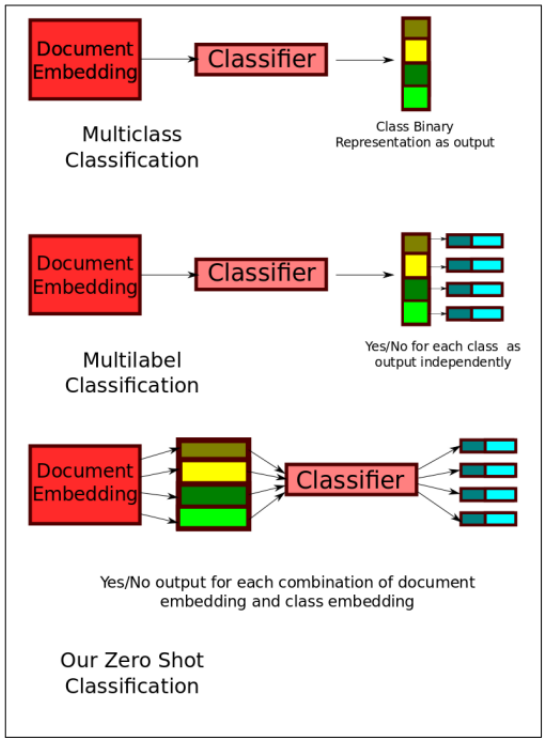
[3]

En este trabajo se propusieron muchos enfoques de zero-shot Learning en el dominio de la visión computacional por Sandouk Chen (2016), Socher et al. (2013). Sin embargo, existe una cantidad muy limitada de trabajos sobre zero-shot Learning en el dominio del procesamiento del lenguaje natural o NLP. En el año 2016 que fue publicado este es el primer trabajo que reporta una solución de zero-shot Learning para la categorización de texto

II-C1. Metodo: En este trabajo se sugirieron muchos enfoques de zero-shot Learning en el dominio de la visión computacional por Sandouk Chen (2016), Socher et al. (2013). Sin embargo, existe una cantidad muy limitada de trabajos sobre zero-shot Learning en el dominio del procesamiento del lenguaje natural o NLP. En el año 2016 que fue publicado este es el primer trabajo que reporta una solución de zero-shot Learning para la categorización de texto. La arquitectura que se preenta es una red neuronal de una sola capa en la concatenación de 1. La inserción media de la oración y 2. La inserción de la etiqueta. Está inspirado en arquitecturas superficiales que obtienen buenos puntajes en tareas de clasificación de texto como Joulin et al. (2016). La segunda arquitectura, en lugar de tomar una media de incrustaciones antes de pasarla a la capa de clasificación, intenta modelar la secuencia utilizando un LSTM Hochreiter Urgen Schmidhuber (1997). Nuestra

tercera arquitectura LSTM puede considerarse similar a la arquitectura usada por Wang et al. (2016) para el análisis de sentimientos basado en aspectos. En lugar del "Término de aspecto", pasamos la inserción de la etiqueta para que se considere relacionada

II-C2. Conclusión: En este trabajo, se presentaron técnicas y modelos que se pueden usar para la clasificación de Zero-Shot learning en textos. Se prueba que los modelos pueden ser mejores que las precisiones de clasificación aleatoria en conjuntos de datos sin ver ni un ejemplo. Se puede decir que esta técnica aprende el concepto de relación entre una oración y una palabra que pueden extenderse más allá de los conjuntos de datos. A través de esto los niveles de precisión dejan mucho margen para trabajos futuros.



[3]

Figura 2. Arquitectura de clasificación multiclase, multietiqueda y clasificación zero-shot learning propuesto

III. Revisión de la Literatura

Se han buscado trabajos relacionados en páginas como Scopus y Arxiv. Se realizaron las siguientes consultas:

III-A. Cadenas de búsqueda

Se han buscado trabajos relacionados en páginas como Scopus y Arxiv. Se realizaron las siguientes consultas:

- TITLE-ABS-KEY(Zero-Shot Learning) AND (images)
- TITLE-ABS-KEY(Zero-Shot Learning) AND (words)
- TITLE-ABS-KEY(Zero-Shot Learning) AND (Image classification)

Las preguntas sobre las que nos basamos para filtrar nuestros resultados fueron:

- ¿Como empezar la clasificación de imagenes con zero-shot Learning?
- ¿Que metodos usar para la calsifación de imagenes con zero-shot Learning ?

III-B. Resultados: Clasificación de Imagenes

Cuando se lleva a cabo la clasificación de Zero-shot Learning, las clases de prueba incluyen tanto las clases vistas como las no vistas,$X = X_s \cup X_u$ En este momento, las características visuales $f_\phi(x)$ seran expresadas como $g_\phi(x)$ en un espacio en comun. Las características semánticas asignadas a la los espacios comunes son $h_\phi(v)$, donde $v = v_c \cup v_d$. El grado de coincidencia de características visuales y características semánticas de la imagen, es decir, la puntuación de similitud s, se puede calcular se la siguiente manera:

$$s = r_\omega(C(g_\phi(x), h_\phi(v)))$$

La clase con la puntuación de similitud más alta se toma como etiqueta de la predicción. Esta expresada como:

$$Y = argamax(s)$$

[7]

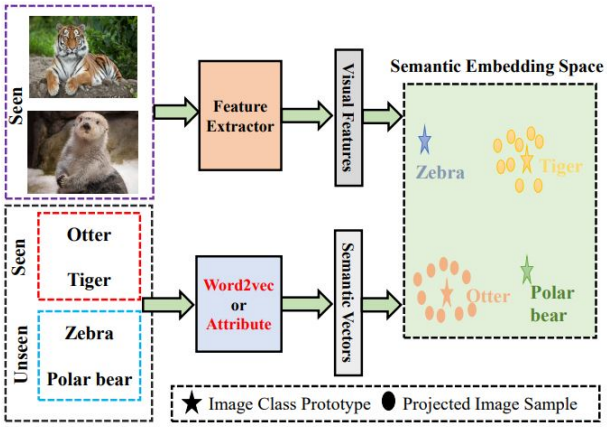
III-C. Resultados: Metodos GZSL Inductivos y Semanticos

La idea principal de GZSL es clasificar objetos de ambos dominios (clases vistas y no vistas) transfiriendo el conocimiento de las clases vistas a las invisibles a través de representaciones semánticas. Para lograr esto, se deben abordar dos cuestiones clave:

- Cómo transferir el conocimiento de las clases vistas a las invisibles
- Cómo aprender un modelo para reconocer imágenes de clases visibles y no vistas sin tener acceso a las muestras etiquetadas de clases invisibles

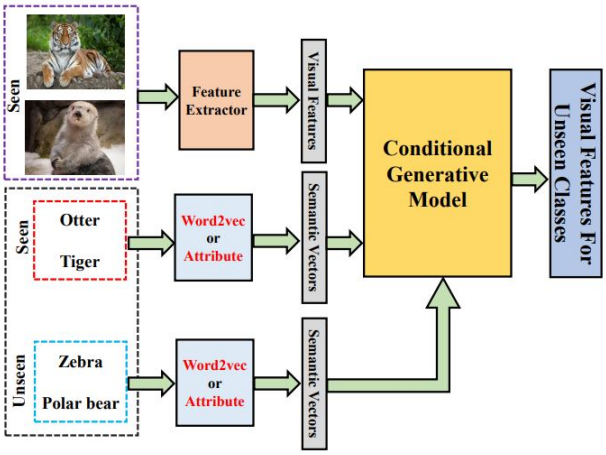
Al respecto, se han propuesto muchos métodos, que pueden clasificarse ampliamente en métodos basados en incrustación y basados en generación.

- **Embedding-based Metodos:** Aprender una función de proyección o incrustación para asociar las características visuales de bajo nivel de las clases vistas con sus correspondientes vectores semánticos. Se utiliza la función de proyección aprendida
  Reconocer clases novedosas midiendo el nivel de similitud entre las representaciones de prototipos y las representaciones predichas de las muestras de datos en el espacio de incrustación.
- **Generative-based Metodos:** Aprender una función de proyección o incrustación para asociar las características visuales de bajo nivel de las clases vistas con sus correspondientes vectores semánticos. Se utiliza la función de proyección aprendida
  Reconocer clases novedosas midiendo el nivel de



[2]

similitud entre las representaciones de prototipos y las representaciones predichas de las muestras de datos en el espacio de incrustación.



[2]

IV. Metodología

En esta sección se describen las herramientas y la métodología que se usarán en el presente trabajo

IV-A. Metodología de trabajo

Para implementar este trabajo usaremos el método basado en embedding, para ellos listaremos previamente los 2 tipos de datos(datos de entrenamiento y datos de Zero shot Learning(ZSL)).

No utilizaremos imagenes de ZSL en el entrenamiento, ya que no forma parte del modelo, sin embargo, si necesitamos representar sus clases como datos.

1. Realizaremos un embedding de imagenes para los datos de entrenamiento por medio de una red convolucional, en este caso una red entrenada.
2. Realizaremos un embedding de clases(de los datos de entrenamiento y de ZSL), para ello usaremos word2vec con el objetivo de que la red aprenda a relacionar una entrada con un vector del espacio word2vec.
3. Realizaremos el embedding de la imagen a clasificar por ZSL, para comparar el vector de caracteristicas de la imagen con todos los vectores de clase que tenemos tanto de entrenamiento como de ZSL por medio de una busqueda vecinos mas cercanos.

IV-B. Métricas

En el presente trabajo se evaluará la precisión promedio por clase-top-1, métrica para evaluar el rendimiento de Zero shot learning Es decir encontramos la precisión del reconocimiento para cada clase por separado y luego la promediamos entre todas las clases.

Para un conjunto de clases Y con N clases, la precisión promedio por clase-top-1 viene dada por: [6]

$$a_y = \frac{1}{N}\sum_{c=1}^{N}\frac{\text{número de predicciones correctas}}{\text{número de muestras en C}}$$

IV-C. Pytorch-Transformers

PyTorch-Transformers [5] es una biblioteca de modelos pre-entrenados de última generación para el procesamiento del lenguaje natural (NLP), que ahora se llama Transformers y es desarrollado por HuggingFace.

Esta biblioteca contiene implementaciones de PyTorch, pesos de modelos previamente entrenados, scripts de uso y utilidades de conversión para los siguientes modelos: BERT, GPT, GPT-2 (de OpenAI), Transformer-XL, XLNet, XLM.

IV-D. SimpleTransformers

SimpleTransformers [4] es una libreía construida en base en la biblioteca Pytorch-Trasformers [5], que también contiene modelos de arquitectura Transformer pre-entrenados pero cuyo objetivo principal es simplificar la codificación y evaluación de los modelos.

V. Experimentación y resultados

El código que se desarrollará utilizará el dataset AwA2(Animals with Attributes 2), que comprende imagenes de distintos clase de animales los cuales tienen atributos definidos por cada imagen, el dataset fue extraido de [8]

V-A. Data Set

Consta de 37322 de imágenes 50 clases de animales con representaciones de características extraídas previamente para cada imagen, Los datos de imágenes se recopilaron de fuentes públicas, como Flickr, en 2016.

En este dataset de animales de los 50 tipos de animales estos tienen 85 características cada animal

Solo son imágenes con licencia para uso gratuito. [8]

Figura 3. Dataset de animales
[8]

V-B. *Explicación*

- Por primer paso se hizo un tratamiento al dataset semantico (clase, atributos) unificando las clases con sus respectivos atributos, posterior a ello se hizo la separacion y ordenamiento de datos para el entrenamiento y el zero-shot learning.
- Por segundo paso se hizo el preprocesamiento del dataset visual(imagenes) , redimensionamiento y se separó las imagenes de entrenamiento y de zero-shot learning.
- Por tercer paso se realizó la extraccion de caracteristicas de las imagenes de entrenamiento por medio del modelo de resnet18, separados en minibatches de 4.
- Por cuarto paso se realizó el entrenamiento en una red MLP teniendo como entrada el vector de 512 caracteristicas y como capa de salida los 85 atributos.
- Por quinto paso se debió realizar el test con las imagenes de zero shot learning con el fin de obtener la predicción del vector de atributos.
- Y finalmente ser evaluado por una metrica de distancias para comprobar la precisión del modelo.

VI. Conclusión

- Se aprendió el funcioanmiento del algoritmo Zero-shot learning.
- Se desarrolló código para la implementación de la extraccion de caracteristicas y la funcion de proyeccion del espacio visual al espacio semántico.
- NO se pudo comprobar la precisión del modelo, por una falta de conocimiento en la implementación de este.

Referencias

[1] An embarrassingly simple approach to zero-shot learning ,Bernardino Romera-Paredes,Philip H. S. Torr,University of Oxford, Department of Engineering Science, Parks Road, Oxford, OX1 3PJ, UK Disponible en https://proceedings.mlr.press/v37/romera-paredes15.pdf
[2] A Review of Generalized Zero-Shot Learning Methods.Farhad Pourpanah, Moloud Abdar, Yuxuan Luo, Xinlei Zhou, Ran Wang Chee Peng Lim, and Xi-Zhao Wang Disponible en https://arxiv.org/pdf/2011.08641.pdf
[3] Zero-Shot Learning - The Good, the Bad and the Ugly ,Yongqin Xian, Bernt Schiele, Zeynep Akata Disponible en https://arxiv.org/pdf/1712.05972.pdf
[4] Simple Trnasformers https://simpletransformers.ai/
[5] Pytorch Transformershttps://pytorch.org/hub/huggingface$_pytorch-transformers/$
[6] Zero-shot Learning, https://learnopencv.com/zero-shot-learning-an-introduction/
[7] Zero-Shot Image Classification Based on a Learnable,Deep Metric,Jingyi Liu, Caijuan Shi *, Dongjing Tu, Ze Shi and Yazhi Liu,College of Information Engineering, North China University of Science and Technology https://www.mdpi.com/1424-8220/21/9/3241/pdf
[8] Data-set AwA-2 ,Christoph H. Lampert,Daniel Pucher,Johannes Dostalhttps://cvml.ist.ac.at/AwA2/
[9] Numpy https://numpy.org/doc/stable/user/whatisnumpy.html