

## 1 INTRO

- **Portada** : Hola, soy Raquel y presentaré el trabajo Wordnet y Deep Learning: Una posible Unión. Mis directores son Dario Garcia Gasulla y Ulises García.
- **Estructura** : La estructura del trabajo es muy similar a la de la memoria escrita:
  - Primero hablaré de los conocimientos necesarios para entender el trabajo, con menos detalle que en la memoria a causa de las limitaciones del tiempo.
  - Después explicaré el trabajo relacionado con el análisis posterior.
  - Seguidamente explicaré el enfoque y el análisis.
  - Finalmente comentaré las conclusiones obtenidas.

## 2 REDES NEURONALES

- **Que explicaré**: En este apartado explicaré a grandes rasgos como funcionan redes neuronales, redes convolucionales y transfer learning.
- **Una Neurona**: Una red neuronal es un modelo computacional de aprendizaje inspirado en la forma de procesar la información del cerebro.

Su unidad básica de computación es una neurona. Cada neurona actúa como una función sobre unas ciertas entradas y genera una salida. En la figura podemos ver un ejemplo de neurona. A la izquierda tenemos su entrada, luego los pesos, que se combinan linealmente con la entrada y se transforman utilizando una función de activación no lineal.
- **Por capas**: Las neuronas se suelen organizar en capas, las más comunes son las capas *fully-connected*, como la que se ve en la figura. En ella podemos ver la estructura base de una red neuronal, donde a la izquierda vemos la entrada, en el centro las capas ocultas, acompañadas de sus correspondientes sesgos y finalmente a la derecha la salida. Por cada arista hay un peso, que se entrena calculando las derivadas parciales utilizando backpropagation y luego minimizando un error utilizando el algoritmo de *Stochastic Gradient Descent*.

## 3 REDES CONVOLUCIONALES

Las redes convolucionales son muy similares a las redes neuronales explicadas, las mayores diferencias son que asumen una estructura concreta en la entrada, y contienen neuronas

convolucionales, es decir, aquellas que operan con una convolución, que generalmente opera una función de kernel sobre la imagen, como podemos ver en el ejemplo de la figura. Donde podemos ver a la izquierda la imagen general, en el centro el kernel y a la derecha la convolución resultante.

En redes convolucionales, la disposición por capas permite representar la información más compleja a partir de otra más simples. Por tanto puede acabar extrayendo features abstractas, como las que vemos en la segunda figura.

## 4 TRANSFER LEARNING

Otro concepto importante en este trabajo es el de *transfer learning*, que es el campo de estudio que reutiliza el lenguaje de representación en otro. Puesto a que las redes convolucionales tienen una gran cantidad de parámetros, muchas veces es demasiado costoso a nivel de recursos computacionales o de cantidad de datos para entrenar. Utilizando transfer learning se pueden solventar estos problemas. Comúnmente se suele utilizar para:

- **fine tuning**, que consiste en tomar los pesos ya entrenados por el conjunto de datos de origen, y a continuación, entrenar la red sobre esta inicialización utilizando el conjunto de datos de la tarea de destino.
- O para **Feature Extraction**, que consiste en procesar un conjunto de datos a través de una red neuronal ya entrenada y extraer valores de activación para que puedan ser utilizados por otro mecanismo de aprendizaje. Un ejemplo de estructura básica para feature extraction sería el de la figura.

## 5 FNE

En transfer learning para feature extraction generalmente se fijan todas las capas menos las dos últimas, que se entrenan con el nuevo conjunto de datos. Sin embargo en este trabajo nos centramos en el Full-network embedding, un embedding que, a diferencia del anterior, permite entrenar todas las capas. Creando de esta forma un lenguaje de representación mejor. Éste se obtiene con el siguiente algoritmo:

- Primero se realiza un forward pass del conjunto de datos target por toda la red y se guardan los valores de activación de las neuronas.
- Seguidamente se le aplica un pooling espacial a estos datos.
- A continuación se aplica una estandarización a las features.
- y Finalmente al resultado se le aplica una discretización.

La estructura del proceso sería la que podemos ver en la figura. Donde se extraen las features de todas las capas y se someten al

algoritmo para obtener el embedding final. Que se podría utilizar para la tarea objetivo utilizando algún método, como por ejemplo una SVM.

## 6 WORDNET

Wordnet es una base de datos que contiene nombres, verbos, adjetivos y adverbios en conjuntos de sinónimos (que llamaremos synsets). Los synsets están conectados entre ellos por medio de relaciones conceptuales, semánticas y léxicas. Como serían la de sinonimia, hiponimia e hiperonimia.

## 7 IMAGENET

Imagenet es una base de datos de imágenes organizada utilizando la jerarquía de wordnet. Su principal objetivo es dotar a los investigadores en campos relacionados con visión artificial de una base de datos a gran escala con la que poder trabajar.

Para el full-network embedding utilizaron el subconjunto correspondiente al reto de imagenet de 2012 de reconocimiento de imágenes. Éste consta de:

- Un conjunto de datos de entrenamiento de 1.2 millones de muestras y 1,000 categorías diferentes.
- Un conjunto de datos de validación de 50,000 muestras y 1,000 categorías.

## 8 DATOS INICIALES

En nuestro caso estudiaremos un full-network embedding obtenido a partir de la tarea origen (T S = Clasificación de imágenes, D S = 1.2 M imágenes de Imagenet con sus correspondientes clases) y una tarea de destino (T T = Clasificación de imágenes, D T = 50,000 imágenes de Imagenet con sus correspondientes clases).

Es decir, partiendo de una red convolucional entrenada con todo el conjunto de datos de entrenamiento de Imagenet, hemos generado un lenguaje de representación para el conjunto de datos de validación de Imagenet. Lo que resulta en el full-network embedding con el que trabajaremos. El full-network embedding consiste en una matriz de tamaño 50,000 muestras por 12,416 características, como observamos en la figura.

Respecto a los *synsets* que analizaremos, hemos elegido los siguientes por estar distribuidos de una forma bastante uniforme entre las clases, y por ser de dos "ramas" lo más diferentes posibles, además representan niveles de abstracción parecidos.

## 9 OBJETIVOS

Los objetivos del análisis que haremos son:

- Analizar el *embedding* dado y el comportamiento de las *features* en las distintas capas.
- Analizar si hay alguna relación entre el *embedding* y los *synsets* seleccionados.

## 10 ESTADÍSTICAS

## 11 HIPÓTESIS

## 12 DE WN A FNE

## 13 SYNSETS

## 14 DE FNE A WN

## 15 CONCLUSIONES