

1. Problema

El presente trabajo pretende comparar el performance para la clasificación de imágenes con dos técnicas distintas en diversas bases de datos, así como la reconstrucción de imágenes y la prueba con bases completamente distintas. Los resultados y código para este trabajo se encuentran el siguiente [Github](#).

El uso de redes convolucionales ha sido de los más populares para la tarea de visión computacional, sin embargo, recibe fuertes críticas sobre todo desde un lugar filosófico, donde la forma en la que están aprendiendo a ver no se asemeja a la idea que se tiene del como aprendemos los humanos, por ello Hinton en [2] propone una alternativa que es nombrada como cápsulas y cuya idea es dejar que las redes aprendan más allá de solo líneas y formas, para aprender también sobre posición y rotaciones(que pareciera ser la forma en la que aprendemos).

Sabour, Frosst y Hinton en [1](2017) proponen la primer forma para poder entrenar una cápsula[2](2011). Una idea innovadora que pretende combatir con las muy conocidas Redes Convolucionales(CNN). Por ello este trabajo tiene como finalidad comparar el performance de estos dos tipos de redes.

2. Resultados y discusión

Para la implementación de CapsNet nos basamos del código en [3] el cual tiene implementada la opción para hacer data augmentation (width/height_shift_range). Para todas las bases de datos se consideraron las variantes utilizando el shift = 0.49 y el no aumentar los datos, es decir, shift = 0. Para este modelo se requiere tener las imágenes en escala de grises y de un tamaño de 28x28 por lo cual, se realizaron los cambios necesarios. La estructura del modelo fue la planteada en [1].

En cuanto a la implementación del modelo CNN utilizamos la arquitectura mobilenet versión 2, de igual forma se rescalaron a grises las imágenes para hacer de forma justa los entrenamientos. Para este comparativo no ocupamos la transferencia de conocimiento, por lo que solo echaremos mano de la forma de la arquitectura.

En cuanto a replicar el paper original, los resultados son mostrados a lo largo de las siguientes tablas; para el caso de la reconstrucción de las imágenes, solo se presenta en la figura 1 un ejemplo, algunos ejemplos más se encuentran dentro del repositorio antes mencionado.

Para llevar a cabo la comparación entre estos modelos, se seleccionaron 3 bases de datos

distintas: MNIS, FASHIONMNIS y Cifar10. Como ya se había mencionado con anterioridad, en todos los casos se ajusto a imágenes de 28x28 (en caso de ser necesario) y se cambio a escala de grises para poder entrenar CapsNet.

Los resultados de los modelos antes descritos se muestran en la tabla 1. Podemos observar que en el caso de CapsNet, como es de esperarse, el aumentado de datos no genera una gran diferencia respecto a los resultados vistos en las imágenes originales; para el caso de CNN-Mobilnet es claro que las transformaciones tienen un efecto positivo.

Base de datos	Epocas	Aum. datos	CapsNet Acc	CNN Acc
MNIST	50	no	99.7 %	99.2 %
	50	si	99.6 %	99.6 %
Fashion MNIST	50	no	91.5 %	91.2 %
	50	si	91.2 %	93.7 %
Cifar 10	40	no	55.3 %	71.2 %
	40	si	56.7 %	72.6 %

Cuadro 1: Resultados para el modelo CapsNet vs CNN en distintas bases de datos y entrenados en un número similar de épocas. La variable de interes en este caso es el accuracy obtenido para cada una de las combinaciones presentadas

En orden de poder explorar más acerca de los resultados ya obtenidos, se presenta en la tabla 2 un comparativo de los modelos para MNIST observados en la tabla 1, utilizando como base de prueba affNIST, esta base consta de transformaciones afines a la base MNIST y por ello son datos completamente nuevos para los modelos planteados. La tabla 2 deja ver de forma clara que los modelos de CapsNet son más robustos que los obtenidos por la red CNN en cualquiera de las versiones aquí planteadas.

Modelo	Acc
CAPSNET - sin DA	48.0 %
CAPSNET - DA	50.9 %
mobilnet - Sin DA	26.2 %
mobilnet - DA	38.7 %

Cuadro 2: Resultados de accuracy para los modelos de MNIST tomando como base de prueba affNIST, base compuesta de transformaciones afines a MNIST

En la figura 1 se presenta la reconstrucción de imágenes aplicada al data set FASHION-MNIST. Los primeros cinco renglones representan la imagen verdadera, mientras que los últimos son los obtenidos por el modelo.



Figura 1: Algunos ejemplos para la reconstrucción de imágenes en la base de datos de Fashionmnist. En la parte superior se encuentran las imágenes reales y abajo su correspondiente reconstrucción

3. Conclusión

Si bien podemos resolver el problema filosófico de las redes convolucionales para poder aprender de rotaciones, este no cumple con la idea de enseñar a la máquina a ver como nosotros lo hacemos. Se presentaron diversos escenarios, incluso donde CapsNet presenta un mejor performance usando pequeñas variaciones de los datos, sin embargo, incluso sin usar el aumento de datos CapsNet parece tener un mejor rendimiento aprendiendo de forma conjunta todas las partes de un objeto.

Si bien la ganancia no se ve directamente relacionada en tiempo, en algunos casos si se logra ver en la métrica de interés y lo robusto de los modelos, lo que nos lleva a pensar que quizá con la evolución de las computadoras y la creación de algoritmos más eficientes para entrenar (es decir poder entrenar más épocas en menos tiempo) CapsNet podría convertirse en una red muy competitiva para la visión computacional.

En general el tiempo que tarda en entrenarse una cápsula, es mucho mayor al que le tomó el esquema de una red convolucional, sin embargo, esto lo podemos ver como un estado inicial para la idea de CAPSNET. Existen otras formas de entrenar las cápsulas que han resultado del interés para los investigadores, una de las más recientes es el uso del algoritmo EM.

Referencias

- [1] Sabour, Sara & Frosst, Nicholas & Hinton, Geoffrey. (2017). Dynamic Routing Between Capsules.
- [2] Geoffrey E Hinton, Alex Krizhevsky, & Sida D Wang. Transforming auto-encoders. In International Conference on Artificial Neural Networks, pages 44–51. Springer, 2011.
- [3] A Tensorflow implementation of CapsNet(Capsules Net) in Hinton's paper Dynamic Routing Between Capsules [Github-naturomics](#)

Apéndice

El material con el que se realizó este trabajo se encuentra en el siguiente repositorio [Github](#), así como las instrucciones para que los programas corran de forma correcta