UNIVERSIDAD DE SANTIAGO DE CHILE
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA INFORMÁTICA

Tarea - Redes Neuronales
(Going Deeper with Convolutions)

Integrantes:	Gustavo Aguilar Morita
	Sandra Hernández Yamunaqué
Curso:	Inteligencia Computacional
Programa:	Magíster en Ingeniería Informática
Profesor Cátedra:	Gonzalo Acuña Leiva
Profesor Laboratorio:	Héctor Rojas Pescio

30 de Agosto de 2021

1. Introducción

El paper de estudio nos presenta una arquitectura llamada GoogLeNet la cual es una variante de Inception Network y la cual fue presentada en el desafío ILSVRC14 (ImageNet Large-Scale Visual Recognition Cha-llenge 2014). GoogLeNet es una arquitectura que busca ser una propuesta más eficiente y exacta que sus predecesoras para resolver problemas de clasificación y/o detección de objetos a partir de la inspiración y uso de conceptos como filtros Gabor fijos de diferentes tamaños para manejar múltiples escalas, el enfoque Network-in-Network para aumentar el poder de representación de las redes neuronales asi como redes neuronales convolucionales como Inception de la cual usa varias capas que se repiten a lo largo de la arquitectura de GoogLeNet así como tambien del uso de capas convolucionales de 1x1 que aumentan la profundidad de la red y reducen la dimensionalidad de la misma, aumentando así el ancho de la red sin que exista una penalización significativa del rendimiento además de abaratar los costos computacionales significativamente la cual hace a GoogLeNet una gran alternativa a escoger.

2. Marco Teórico

2.1. ILSVRC14 (ImageNet Large-Scale Visual Recognition Challenge 2014)

ImageNet es banco de datos de imágenes organizado acorde a la jerarquía de WordNet (bajo el concepto synset o en español conjunto de sinónimos), que es utilizado para entrenar modelos de reconocimiento de objetos a gran escala. Se considera un banco de datos difícil de abordar dada la variedad y complejidad de las imágenes (imágenes borrosas, de diferentes calidades, con ruido, etc..) tanto es así que es considerado como una buena métrica o manera de medir el avance en el campo de la visión artificial. Cabe mencionar que el ganador del desafío de reconocimiento visual a gran escala de Imagenet del año 2014 fue GoogLeNet (ImageNet).

2.2. GoogLeNet

Red neuronal convolucional profunda de 22 capas, variante de Inception Network, que resuelve el problema de sobreajuste que se produce al aumentar la cantidad de capas y unidades dentro de una red neuronal así como los costos asociados. Lo mencionado anteriormente se logra mediante la utilización de los módulos Inception presentes en la arquitectura, dichos módulos detectan características en diferentes escalas a través de convoluciones con diferentes filtros que además reduce el costo computacional de entrenar una red extensa a través de la reducción dimensional que generan las capas de 1x1 utilizadas en varios módulos de la red.

2.2.1. Módulo Inception

La arquitectura Inception tiene el objetivo de encontrar la estructura local óptima y repetirla espacialmente, para ello se construye capa a capa analizando las estadísticas de correlación de cada capa, agrupando aquellas unidades con altos valores de correlación, lo anterior sucede de manera secuencial, es decir, las agrupaciones formadas son las unidades de la capa siguiente. Luego el módulo Inception es simplemente la concatenación de 3 escalas

de convolución una de 1x1, 3x3 y de 5x5, sin olvidar la capa de agrupación de 3x3 como se puede observar el la figura 1. Las convoluciones de 3x3 y de 5x5 resultan ser muy costosas computacionalmente y es por ello que resulta necesario reducir la dimensionalidad de las mismas usando convoluciones de 1x1 las cuales a su vez son usadas como función de activación de unidad lineal rectificada como se obversa en la figura 2. Lo mencionado anteriormente se basa en el enfoque Network-in-Network de Lin en (5).

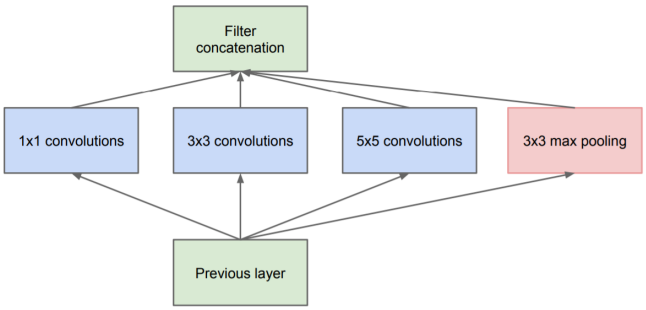

Figura 1: Inception module, versión naive.

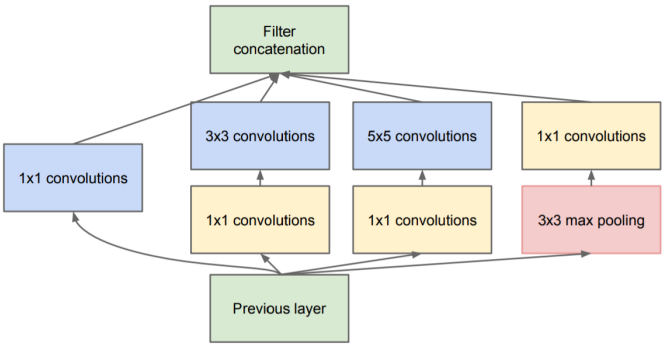

Figura 2: Inception module con reducción de la dimensionalidad.

3. Hallazgos

Dentro de los algoritmos de aprendizaje profundo, en la literatura podemos encontrar diversas arquitecturas que definen un diseño de redes neurales convolucionales de la que se puede evidenciar en el paper (9) y en particular el modelo de *GoogleNet*, cuyo inception module, permite un aprendizaje utilizando filtros de diferentes tamaños (1x1, 3x3 y 5x5), así como también aplicando max pooling de 3x3, y que finalmente se constituye en una concatenación, como anteriormente se ha señalado e ilustradas en las figuras 1 y 2, que posteriormente puede ser utilizado en una siguiente capa.

Ahora bien, la importancia de este modelo constituye una estrategia bastante interesante para poder resolver, por ejemplo, el análisis de imagenes que como bien se sabe, existen diversos factores que alteran el procesamiento, donde tenemos variación de resoluciones, tamaños, o que algunas tengan aplicado algún tipo de filtro, entre otros aspectos, que aumentan la dificultad o el desempeño al momento de generar una predicción.

El problema general de poder mejorar el rendimiento de una red neuronal profunda, podría ser incrementar su tamaño, esto es aumentar la profundidad de las capas y/o incrementar el número de parámetros, sin embargo esto está limitado por los recursos computacionales, que de acuerdo (8), existen diversas técnicas de reutilización de la memoria que permiten reducir costo computacional al momento de entrenar una red, por lo que queda en evidencia la importancia de escoger un diseño apropiado con el fin de maximizar los recursos disponibles.

Por ejemplo, de acuerdo a la figura 1, consideremos la convolución de 5x5 con 32 canales y un input de tamaño 28x28 con 192 canales. Al aplicar la transformación, se obtendrá un output de 28x28 con 32 canales, como se ilustra en la figura 4.

De esto se puede obtener el costo de esta operación convolucional, como se puede ver en (1), lo que da el orden de ~ 120 millones de operaciones.

$$28 \cdot 28 \cdot 192 \cdot 5 \cdot 5 \cdot 32 = 120{,}422{,}400 \tag{1}$$

Ahora si aplicamos un paso intermedio, como se ilustra en la figura 2, y consideramos una convolución de tamaño 1x1 con 16 canales, entonces generaremos un output de 28x28 con

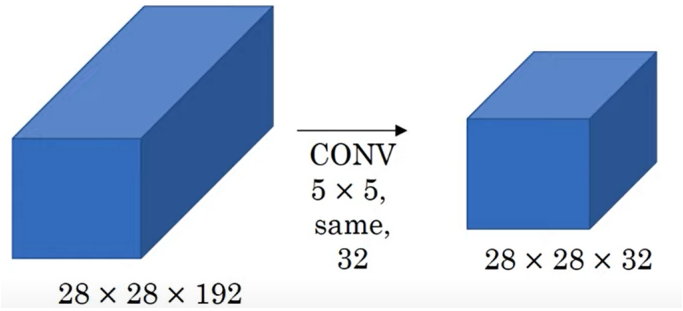

Figura 3: Ejemplo de una convolución de 5x5x32 para input de 28x28x192, cuyo resultado da un costo de $\sim$ 120 millones de operaciones.

16 canales, que posteriormente se utilizará la misma convolución de de la figura 4 tal que el output sea de las mismas características, es decir un tamaño de 28x28 con 32 canales.

$$28 \cdot 28 \cdot 192 \cdot 1 \cdot 1 \cdot 16 = 2{,}408{,}448 \tag{2}$$

$$5 \cdot 5 \cdot 16 \cdot 28 \cdot 28 \cdot 32 = 10{,}035{,}200 \tag{3}$$

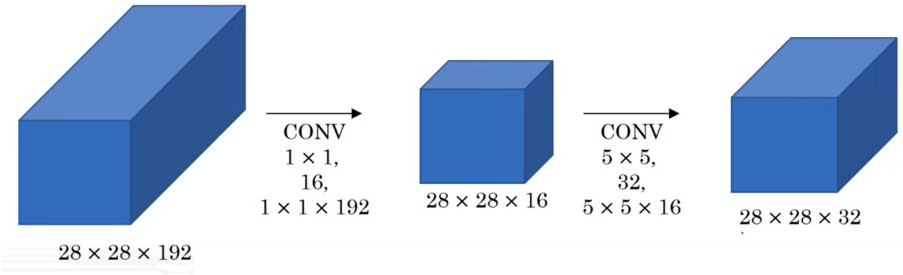

Figura 4: Ejemplo de una convolución intermedia de 1x1x16 y una convolución de 5x5x32, para input de 28x28x192, cuyo resultado da un costo de $\sim$ 12 millones de operaciones.

Calculando estas transformaciones tendremos un total de $\sim$ 12 millones (2 + 3) de operaciones al aplicar una convolución intermedia de 1x1x16, lo que reduce significativamente el costo computacional en comparación a lo obtenido por el modelo naive. De esto último se desprende que la correcta implementación de estos bloques de inception, podrían contribuir de forma importante, en el rendimiento del modelo y así como también en la calidad de la clasificación.

GoogleNet, posee una red de 22 capas en la que se basa sobre la estructura de Network in Network (NIN), que en (7), demuestra que un modelo de capas completamente conectada, conduce al overfitting, y que en consecuencia está fuertemente ligado al método de *dropout*. Se encuentra además que al cambiar de capas completamente conectadas a un average pooling, la precisión aumenta en un 0,6 %.

En el paper de (9) se ilustra la tabla de una arquitectura modelo con las diferentes capas. En ella, se observa que al inicio se aplica una convolución de 7x7 con el fin de reducir drásticamente la dimensionalidad, pero sin perder mayormente la información espacial de la imagen al incorporar un tamaño considerable de filtros.

Un detalle importante es que antes de aplicar un bloque de inception, se aplica un max pooling, esto con el fin de reducir el tamaño de la muestra a través de la red, que de acuerdo a (1) y (10) se evidencia que es un método efectivo para reducir la carga computacional de una red.

Otras investigaciones que han utilizado esta técnica, han tenido importantes resultados en el reconocimiento de imágenes, como en el trabajo de (6), en la que se alcanzó un 80 % en la tasa reconocimiento de brotes de papas en el marco de la clasificación, adaptándose a las condiciones de los investigadores, respecto a sus limitantes computacionales, donde en comparación a otros modelo de CNN, tales como AlexNet y VGG, GoogleNet requiere menos parámetros de entrenamiento.

Sin embargo, otros estudios evidencian que GoogleNet no siempre obtiene los mejores resultados como el trabajo de (4), donde la clasificación anatómica del hígado, riñones, pulmones, espina dorsal, entre otras, se obtuvo que el método CNN de arquitectura de AlexNet fue el que arrojó mejor precisión, donde se indica además que no necesariamente que aumentar el número de capas, aumentaría el rendimiento cuyo proppósito sea mejorar la precisión.

Para entrenar la red GoogleNet, se utilizó el el framework de *DistBelief*, cuya principal característica es evitar el uso de GPU´s, por razones como:

1. Costo, sus valores demasiado altos a la fecha.

2. La mayoría de los GPU's pueden almacenar una cantidad pequeña de data en memoria.

3. Para poder manipular estos componentes requiere de un lenguaje de bajo nivel como C, lo que muchos programadores no tienen conocimientos de estos.

El método de entrenamiento de imágenes es bastante evolutivo o cambiante, según (9), por lo que se hace bastante díficil otorgar una regla maestra que implique la mejor eficiencia en la red. Sin embargo, en el paper de (2), se encuentra que incorporando ciertas transformaciones de imágenes (como manipular cortes o colores) al conjunto de entrenamiento, las predicciones mejoran significativamente, reduciendo incluso la posibilidad del overfitting.

3.1. Sobre la competencia

Los modelos presentados en la competencia de ILSVRC 2014, arrojaron resultados de importante diferencia con respecto a las competiciones anteriores, donde para el año 2012 el top-5 error se encontraba entorno al 15,3 % y para el año 2013 alrededor del $\sim$ 11 %. En el caso de GoogleNet para el año 2014, se obtuvo un valor de top-5 error de un 6,67 %. Este último contiene algunas cosas importantes que destacar, como por ejemplo las 7 versiones de entrenamiento que se generaron, donde solamente se varía los métodos de muestro (aplicando las técnicas descritas anteriormente) y la aleatoriedad de la inputación de las imágenes.

Con respecto al desafío de la detección de imágenes, que consiste en dibujar cajas alrededor de los objetos dentro de las imágenes, donde se considera correcto si existe un match de la clase y que los límites de la caja alcancen una precisión de al menos el 50 %. Los resultados para GoogleNet respecto al índice de mAP (mean average precision), se obtuvo un valor de 43,9 %, lo que significa una diferencia de 3 a 4 puntos respecto a sus contrincantes. Esto se logra incorporando 6 ensables de GoogleNets.

4. Conclusión

El modelo de GoogletNet en el mundo del reconocimiento de imágenes, arroja resultados bastante importantes y que además pueden acoplarse de buena forma a la disponibilidad de los recursos computacionales y sin perder el rendimiento o calidad de los resultados de la clasificación. Este tipo de arquitectura nos otorga una mirada interesante, de cómo influye la incorporación de elementos dentro de la red, como lo son estos bloques de inception, que reducen la dimensionalidad o incluso el muestro de los datos, con el fin de mejorar el costo computacional. Se demuestra empíricamente cómo se reducen las operaciones al incorporar convoluciones intermedias al interior de una capa, lo que conlleva a cuestionarnos, de qué otras formas se podría mejorar, sin perder información o bien sin reducir la capacidad de la clasificación. Ir más profundo no necesariamente signfica mejores resultados, como en la clasificación anatómica presentada en (4), sin embargo para la detección de brotes de papas, se obtuvo buenos resultados en comparación a otras arquitecturas.

Los resultados obtenidos en la competición ILSVRC14, demuestran la solidez de esta metodología, lo que también se logra evidenciar la importancia de implementar una estrategia acertiva en el conjunto de entrenamiento, como la diversificación de las imágenes, utilizando cropping, redimensionamiento, cambio de colores, brillos, entre otros. Además, se observa como aumenta en alrededor de 4 puntos porcentuales el índice mAP cuando se aplica 6 bloques de inception al interior de la red y reduciendo los errores en el modelo, alcanzando un top-5 error de 6,67 %.

Bibliografía

[1] Cotter, S. F. (2020). Mobiexpressnet: A deep learning network for face expression recognition on smart phones. In *2020 IEEE International Conference on Consumer Electronics (ICCE)*, pages 1–4.

[2] Howard, A. G. (2013). Some improvements on deep convolutional neural network based image classification.

[ImageNet] ImageNet. About imagenet.

[4] Khan, S. A. and Yong, S.-P. (2016). An evaluation of convolutional neural nets for medical image anatomy classification. In Soh, P. J., Woo, W. L., Sulaiman, H. A., Othman, M. A., and Saat, M. S., editors, *Advances in Machine Learning and Signal Processing*, pages 293–303, Cham. Springer International Publishing.

[5] Lin, M., Chen, Q., and Yan, S. (2013). Network in network. *arXiv preprint arXiv:1312.4400*.

[6] Ma, J., Rao, J., Qiao, Y., and Liu, W. (2018). Sprouting potato recognition based on deep neural network googlenet. In *2018 IEEE 3rd International Conference on Cloud Computing and Internet of Things (CCIOT)*, pages 502–505.

[7] Min Lin, Qiang Chen, S. Y. (2014). Network in network.

[8] Shirahata, K., Tomita, Y., and Ike, A. (2016). Memory reduction method for deep neural network training. In *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6.

[9] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9.

[10] Zhou, D.-X. (2020). Theory of deep convolutional neural networks: Downsampling. *Neural Networks*, 124:319–327.