

Tarea 2

Redes Neuronales Convolucionales

Integrantes: Vincko Fabres
Profesor: Pablo Estevez.
Auxiliar: Ignacio Reyes Jainaga
Ayudantes: Andrés González
Bastían Andreas
Daniel Baeza
Francisca Cona
Javier Molina
Óscar Pimentel
Pablo Montero
Roberto Cholaky

Fecha de entrega: Sábado 16 de Octubre de 2021
Santiago, Chile

1. Parte teórica

1.1. Tamaños y formas de los parámetros y activaciones

Se debe demostrar que el modelo VGGNet con arquitectura 'D' posee 138 millones de parámetros.

Lo primero a notar es que al ser input una imagen RGB posee 3 canales de entrada.

- En la primera capa:
conv3-64: $(3*3+1)*3*64 = 1920$
conv3-64: $(3*3+1)*64*64 = 4096$
- En la segunda capa:
conv3-128: $(3*3+1)*64*128 = 81920$
conv3-128: $(3*3+1)*128*128 = 163840$
- En la tercera capa:
conv3-256: $(3*3+1)*128*256 = 327680$
conv3-256: $(3*3+1)*256*256 = 655360$
conv3-256: $(3*3+1)*256*256 = 655360$
- En la cuarta capa:
conv3-512: $(3*3+1)*256*512 = 1310720$
conv3-512: $(3*3+1)*512*512 = 2621440$
conv3-512: $(3*3+1)*512*512 = 2621440$
- En la quinta capa:
conv3-512: $(3*3+1)*512*512 = 2621440$
conv3-512: $(3*3+1)*512*512 = 2621440$
conv3-512: $(3*3+1)*512*512 = 2621440$
- Capas fully connected :
fc-4096: $(7*7)*512*4096 + 4096 = 102764544$
fc-4096: $4096*4096 + 4096 = 16781312$
fc-1000: $4096*1000 + 4096 = 4100096$

Siendo el total de parámetros la suma de parámetros por capa, es decir, 139708288 parámetros, los cuales en memoria cada uno usan 32 bits o 4 bytes, es decir, la memoria de parámetros es igual a 558833152 bytes o 532.9 MB de memoria.

2. Parte práctica

Disclaimer: Debido a como arroja los plots Tensorboard se hace necesario advertir que la edición por gráfico sería costosa en tiempo, razón por la cual se opta por comentar que todos los gráficos poseen en coordenada x número de iteraciones y en coordenada y la función Accuracy o Loss debidamente señalada en su título.

2.1. Dropout

Se ejecuta el entrenamiento de la red convolucional con arquitectura de 2 capas convolucionales, 2 capas de pooling y 2 capas fully-connected, variando el parámetro *Dropout*, utilizando una vez valor 0.5 y otra valor 0.0, variando así la probabilidad de apagar las unidades.

Al momento de utilizar 0.5 como probabilidad de Dropout los accuracy del modelo fueron 0.78 y 0.65 para los conjuntos de entrenamiento y validación respectivamente, mientras que la función de loss fue de 0.67 y 1.02 respectivamente como es posible visualizar en Figura 1.

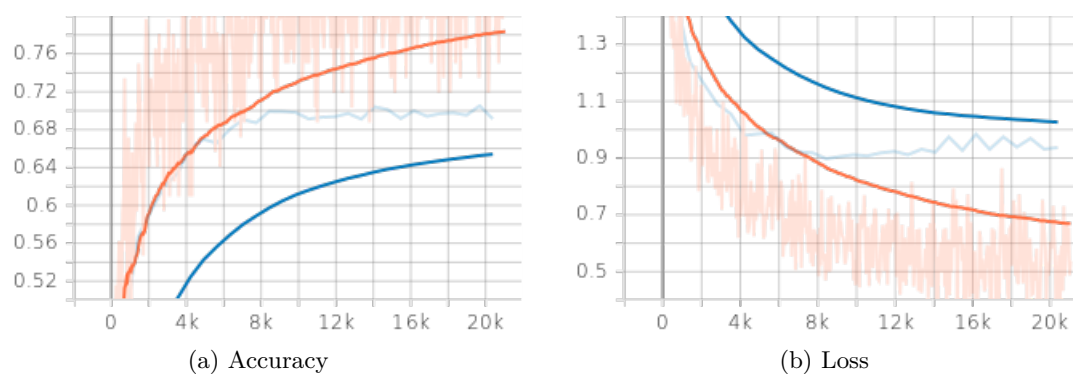


Figura 1: Accuracy y Loss Dropout 0.5

Repitiendo el experimento, esta vez con Dropout 0.0 los accuracy del modelo fueron 0.92 y 0.66 para los conjuntos de entrenamiento y validación respectivamente, mientras que la función de loss fue de 0.26 y 1.21 respectivamente como se aprecia en Figura 2.

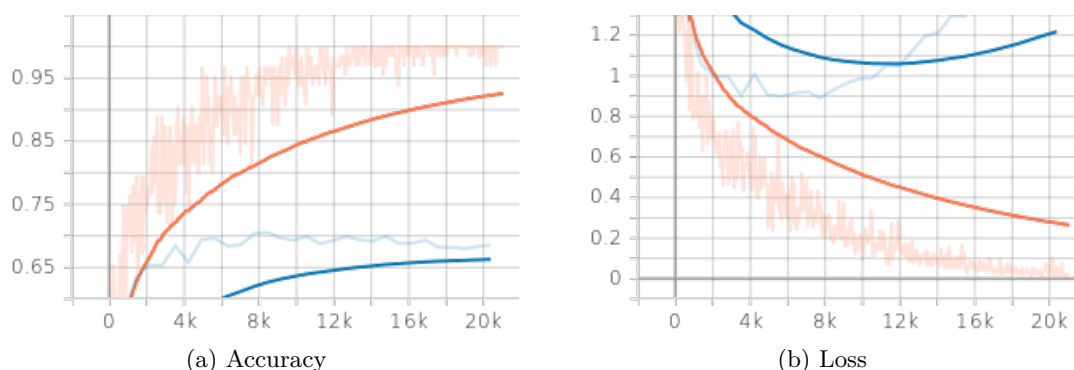


Figura 2: Accuracy y Loss Dropout 0.0

Dado que la diferencia de accuracy es minima y el loss del modelo con Dropout 0.5 es mucho menor se opta por mantener este para las siguientes pruebas, por otra parte, el contraste sugiere que el dropout 0.0 puede tener cierto overfitting.

2.2. Impacto de número de capas

En primera instancia se modifica la red convolucional para que esta vez tenga solo una capa de convolución y pooling, manteniendo el resto de la arquitectura igual. Una vez modificada se debe entrenar para visualizar su rendimiento, retornando los valores para accuracy 0.71 en conjunto train y 0.60 en conjunto de validación y para el loss 0.89 en train y 1.17 en validación, como se puede ver en Figura 3.

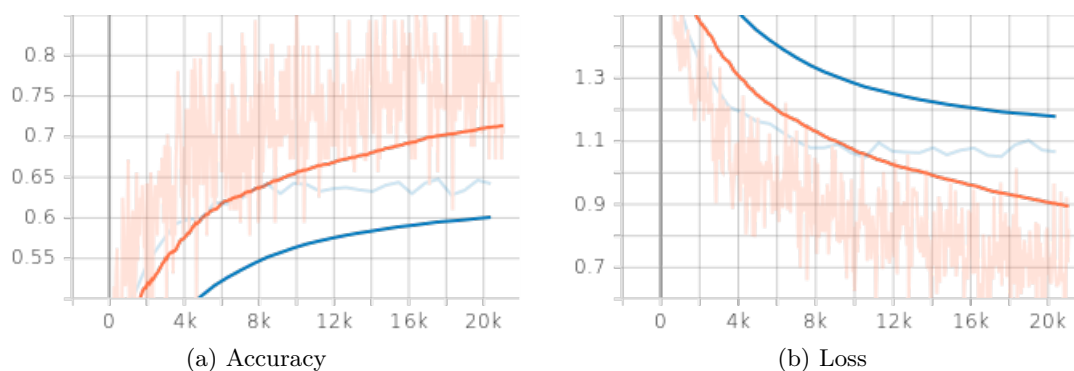


Figura 3: Accuracy y Loss con una capa

Luego se modifica la arquitectura de la red, cambiando el número de capas de convolución y pooling a 3, de igual forma, manteniendo el resto del modelo. Posterior al entrenamiento, la evaluación de desempeño arroja los resultados de accuracy 0.85 y 0.68 para los conjuntos de train y validación respectivamente, en tanto, para la función de loss 0.46 y 0.99 para los mismos conjuntos, como se aprecia a continuación en Figura 4.

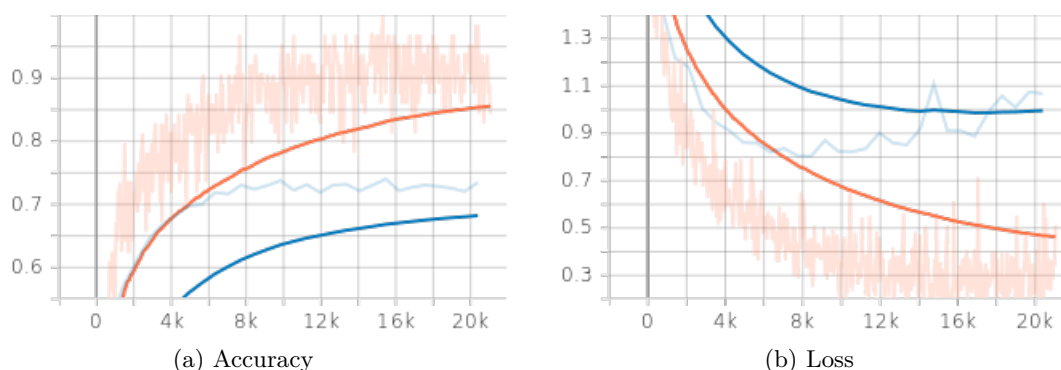


Figura 4: Accuracy y Loss con tres capas

2.3. Comparación con MLP

Se entrena un perceptrón multicapa, el cual posee 100 neuronas en su capa oculta, obteniendo los siguientes resultados; 0.66 y 0.48 en accuracy y 1.04 y 1.49 en loss para los conjuntos de entrenamiento y validación respectivamente, los cuales se visualizan en Figura 5.

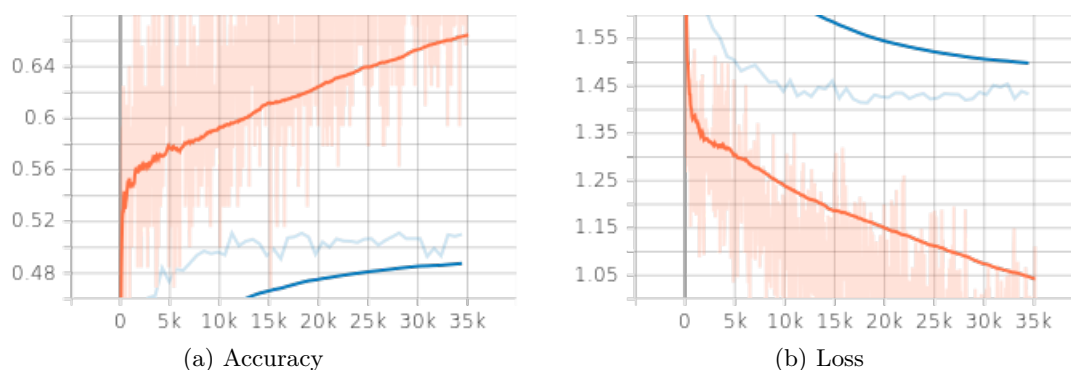


Figura 5: Accuracy y Loss de red MLP

Contrastando con las redes anteriormente creadas la diferencia de tiempos es indistinguible, siendo los entrenamientos de las redes CNN como del MLP en ordenes de entrenamiento de 2 a 3 minutos. En términos de clasificación es donde radica la mayor diferencia, notando claramente el desempeño de ambas en las curvas de accuracy y loss, donde las CNN poseen mejor rendimiento, tanto en términos de clasificación como generalización.

2.4. Data Augmentation

Tomando la mejor red convolucional, es decir, la de arquitectura con 3 capas de convolución y pooling se activa *data augmentation* y se entrena la red, obteniendo esta vez los siguientes resultados luego de 29 épocas de entrenamiento; 0.81 y 0.68 para accuracy en los conjuntos de training y validación respectivamente y en loss 0.56 y 0.94, como es posible apreciar en Figura 6.

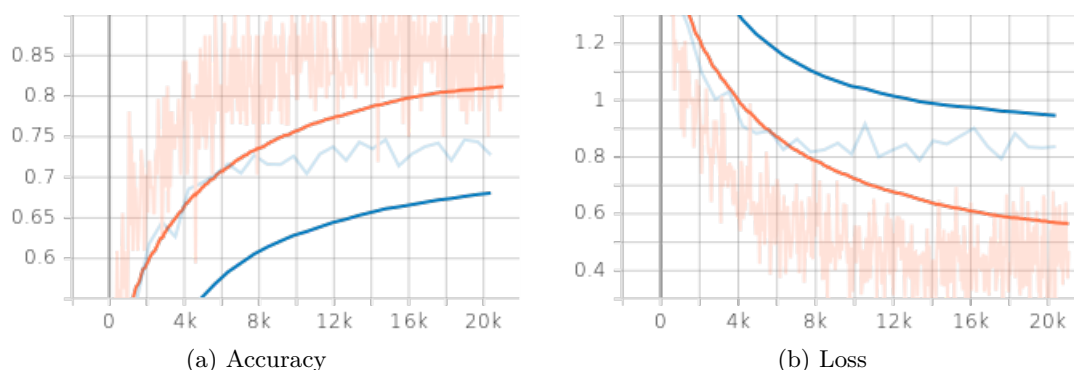


Figura 6: Accuracy y Loss de red de 3 capas con Data Augmentation

Al contrastar con la versión sin data augmentation el número de épocas se mantiene igual, el accuracy para el conjunto de validación es el mismo, no así el de entrenamiento, en el cual la versión de data augmentation posee un menor valor, lo que sugiere que esta red posee menor memorización sobre los ejemplos y por lo tanto generaliza mejor.

3. Parte de programación

El objetivo es construir una red convolucional sin capas *fully connected*, modificando la red convolucional.

La nueva red al no poseer capas *fully connected* requiere una nueva construcción para su vector de salida. Para reducir la dimensionalidad ewspacial de los mapas de características se calcula un promedio por mapa, dando así un vector de salida de tamaño 10.

Los resultados de esta nueva arquitectura son para el conjunto de entrenamiento: 0.75 en accuracy y 0.90 en loss, mientras que para validación: 0.63 y 1.12 para accuracy y loss respectivamente, como se ve en Figura 7.

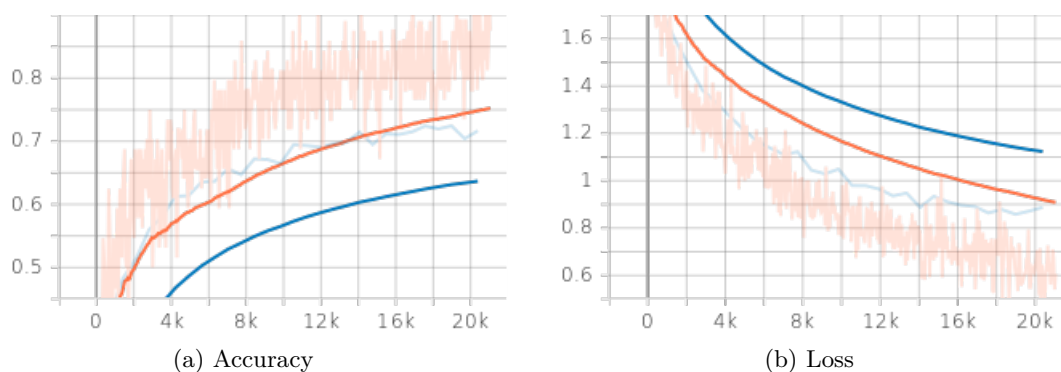


Figura 7: Accuracy y Loss de red de 3 capas sin fully connected

La diferencia entre los accuracy de ambas arquitecturas difieren en un puntaje de 0.05 por lo que

su desempeño es equiparable, con un loss mayor y una clasificación con un rendimiento algo peor, a costa de menor número de parametros, ya que para la red sin fully connected el número de parámetros es de 25000 aproximadamente, en cambio, la que posee fully connected consta de 37000 parámetros, por lo cual el espacio en memoria es considerablemente mayor, siendo para la arquitectura sin fully connected un aproximado de uso de 100000 bytes, por otra parte al incluir estas capas aumenta a 148000 bytes, es decir, un 50 % más de memoria.