Tarea 2 EL4106 MLP

Joaquín Zepeda

October 16, 2021

1 Parte Teórica

A continuación se presenta la parte teórica de la tarea 2, en la cual se presentan los cálculos basados en el modelo de la figura 7 (ver anexo).

Calculemos los parámetros por capas

VGG-16 utiliza 5 bloques de 2-2-3-3-3 de capas convolucionales, seguidos de capas max pooling y 3 capas fully connected.

Se usan filtros de 3×3 en las capas convolucionales:

Conv1 → Corresponde a 2 filtros en cascada de 3×3

pesos del primero: $((3\times3)\times3)\times64 + 64 = 1792$ (3: canales del input; 64: canales del output; 64: bias)

pesos del segundo conectado en cascada: $((3\times3)\times64)\times64 + 64 = 36928$ (3×3: tamaño del filtro)

parametros Conv1 = 1792 + 36928 = **38720** parámetros.

Conv2 → Corresponde a 2 filtros en cascada de 3×3

pesos del primero: $((3\times3)\times64)\times128 + 128 = 73856$ (64: canales del input; 128: canales del output; 128: bias)

pesos del segundo conectado en cascada: $((3\times3)\times128)\times128 + 128 = 147584$ (3×3: tamaño del filtro)

parametros Conv2 = **221440** parámetros.

Conv3 → Corresponde a 3 filtros en cascada de 3×3

pesos del primero: $((3\times3)\times\underbrace{128}_{\text{canales del input}})\times\underbrace{256}_{\text{Canales del output}}+\underbrace{256}_{\text{bias}}=295168$

pesos del segundo conectado en cascada: $((\underbrace{3\times3}_{\text{tamaño del filtro}})\times256)\times256+256=590080$

pesos del tercero conectado en cascada: $((\underbrace{3\times3}_{\text{tamaño del filtro}})\times256)\times256+256=590080$

parámetros Conv3 = 1475328 parámetros.

Conv4 → Corresponde a 3 filtros en cascada de 3×3

pesos del primero: $((3\times3)\times\underbrace{256}_{\text{canales del input}})\times\underbrace{512}_{\text{Canales del output}}+\underbrace{512}_{\text{bias}}=1180160$

pesos del segundo conectado en cascada: $((\underbrace{3\times3}_{\text{tamaño del filtro}})\times512)\times512+512=2359808$

pesos del tercero conectado en cascada: $((\underbrace{3\times3}_{\text{tamaño del filtro}})\times512)\times512+512=2359808$

parametros Conv4 = 5899776 parámetros.

Conv5 → Corresponde a 3 filtros en cascada de 3×3

pesos del primero: $((3\times3)\times\underbrace{512}_{\text{canales del input}})\times\underbrace{512}_{\text{Canales del output}}+\underbrace{512}_{\text{bias}}=2359808$

pesos del segundo conectado en cascada: $((\underbrace{3\times3}_{\text{tamaño del filtro}})\times512)\times512+512=2359808$

pesos del tercero conectado en cascada: $((\underbrace{3\times3}_{\text{tamaño del filtro}})\times512)\times512+512=2359808$

parametros Conv5 = 7079424 parámetros.

Para las capas fully connected:

fc1 : $\underbrace{(7\times7\times512)\times4096}_{\text{Pesos.}}+\underbrace{4096}_{\text{bias}}=102764544$

fc2 : [1×1×4096]

=> $4096\times4096+\underbrace{4096}_{\text{bias}}=16781312$

fc_3 : $[1 \times 1 \times 1000]$

$4096 \times 1000 + \underbrace{1000}_{\text{bias}} = 4.097.000$

Sumando los parámetros de todas las capas:

Total de parametros: 138.357.544

$\approx$ 138 Millones de parámetros.

→ Calculo de memoria en cada capa para almacenar los pesos:
Notemos que las capas de pooling no tienen pesos y tampoco las Imagenes ▼▼

✳ 1 peso ocupa 32 bits (4 bytes) de memoria.

A partir de esto se realizan los calculos.

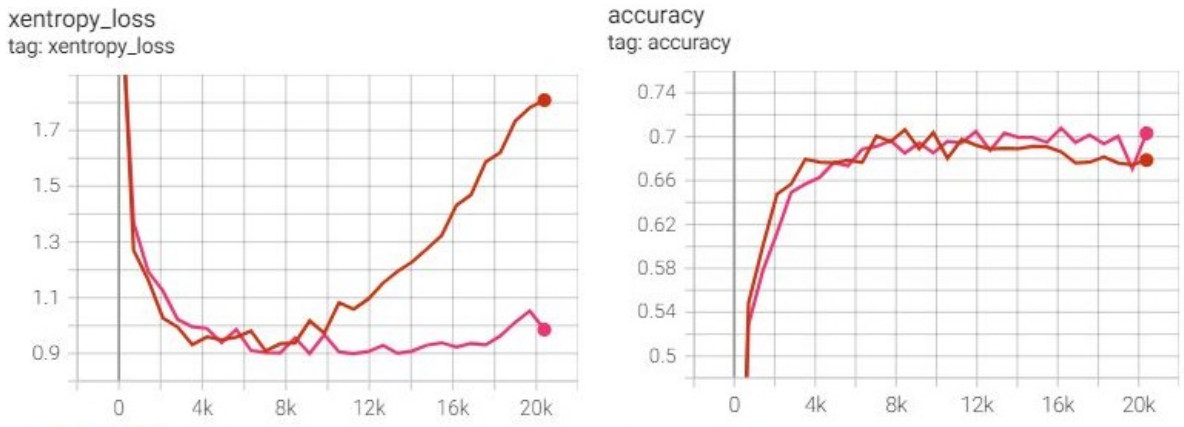
capa	N° Parámetros	memoria (bits)
Conv1	38.720	1.239.040
Conv2	221.440	7.086.080
conv3	1.475.328	47.210.496
Conv4	5.899776	188.792.832
conv5	7.079.424	226.541.568
f_1	102.764.544	3.288.465.408
f_2	16.781.312	537.001.984
f_3	4097 000	131.104.000
TOTAL	138.357.544	4.427.441.408

CORRESPONDE a 553.43 Megabytes.

2 Parte Práctica

2.1 Dropout: utilizando 2 capas convolucionales

A partir de los resultados obtenidos, se puede observar que con la probabilidad de Dropout correspondiente a 0.0 obtiene mejores resultados en el conjunto de validación pero este llega a un valor mucho más alto de error que en comparación con el modelo utilizando la probabilidad de Dropout de 0.5, esto se puede observar en la curva de aprendizaje de la figura 1 en donde se muestra que la tasa de error del modelo usando Dropout 0.0 es aproximadamente el doble que la tasa de error del modelo usando Dropout 0.5, por otro lado, los valores de best accuracy son bastante similares en ambos modelos (0.7002 y 0.6967 para 0.0 y 0.5 probabilidad de Dropout respectivamente), por lo tanto, el modelo elegido para continuar con el desarrollo de la tarea corresponde a la **probabilidad de Dropout igual a 0.5**, esto pues el error en el conjunto de validación de los datos con probabilidad de Dropout 0.0 es muy alto, indicando un posible sobreajuste (overfitting).



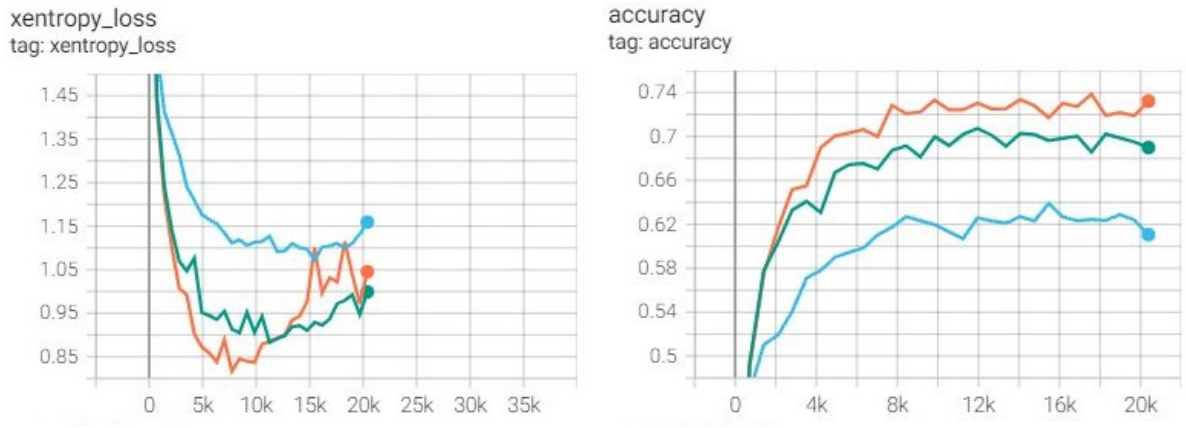
(a) Curvas de aprendizaje de los datos de validación

(b) Evolución del Accuracy de los datos de validación

Figure 1: Gráficos con respecto a los **datos de validación**. En rojo se muestra la red convolucional usando una probabilidad de Dropout de 0.0 y en rosado (magenta) se muestra la red convolucional usando una probabilidad de Dropout de 0.5

2.2 Impacto del número de capas

El número de capas convolucionales en la red neuronal afecta de forma directa a los resultados en **validación**. Como se puede observar en la figura 2, el modelo que tiene una sola capa tiene los peores resultados tanto de la tasa de error, como los resultados en el Accuracy, en donde se encuentra bastante por debajo de los otros dos modelos en donde se utilizan más capas, por otro lado en esta misma figura se puede observar que el modelo que utiliza 3 capas convolucionales obtiene los mejores reusltados de Accuracy pero tiene un error al finalizar el entrenamiento mayor al del modelo utilizando 2 capas convolucionales,a pesar de esto esta capa alcanza el mínimo error de los 3 modelos durante el entrenamiento, y en ese punto ya logra tener mejor Accuracy que los demás, siendo así el mejor modelo, por lo que se selecciona esta configuración para continuar con la tarea.



(a) Curvas de aprendizaje para los distintos números de capas convolucionales

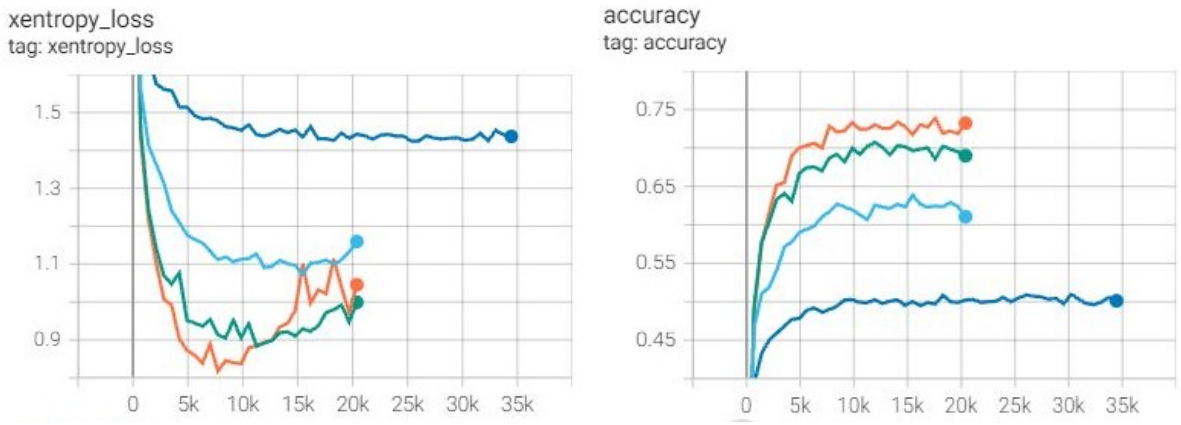
(b) Evolución del Accuracy para los distintos números de capas convolucionales

Figure 2: Comparativa entre los modelos con 1, 2 y 3 capas convolucionales, utilizando los datos de validación.
En celeste se muestran los datos de la red con 1 capa convolucional.
En verde se muestran los datos de la red con 2 capa convolucionales.
En naranja se muestran los datos de la red con 3 capa convolucionales.

2.3 Comparación con MLP

Compare los resultados obtenidos por las distintas redes convolucionales entrenadas y la MLP en términos de error de clasificación, velocidad en el entrenamiento y sobreajuste.

Al comparar los modelos, se puede observar en la figura 3 que en cuanto a los errores de clasificación en los conjuntos de validación, el modelo usando MLP presenta un error considerablemente mayor que los modelos que utilizan capas convolucionales, por otro lado en cuanto al Accuracy de los modelos, la red MLP presenta un menor Accuracy que las redes que utilizan capas convolucionales y por ultimo, al comparar la velocidad de los modelos, la red MLP necesita aproximadamente 15000 iteraciones más que los modelos que utilizan redes convolucionales, por lo que en terminos de velocidad la red MLP es la más lenta. Esto nos indica que para este problema en especifico, las redes MLP con 100 neuronas en la capa oculta no presentan una mejor solución que las redes que utilizan capas convolucionales.



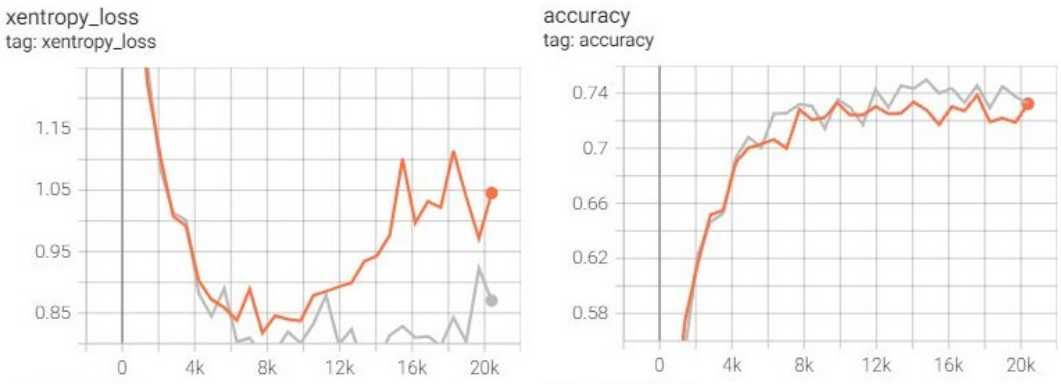
(a) Curvas de aprendizajes

(b) Evolución del Accuracy

Figure 3: Comparativa entre los modelos con 1, 2 y 3 capas convolucionales y el modelo MLP utilizando 100 neuronas en la capa oculta, utilizando los **datos de validación**.
En celeste se muestran los datos de la red con 1 capa convolucional.
En verde se muestran los datos de la red con 2 capa convolucionales.
En naranja se muestran los datos de la red con 3 capa convolucionales.
En azul se muestra la evolución de la red MLP.

2.4 Data Augmentation

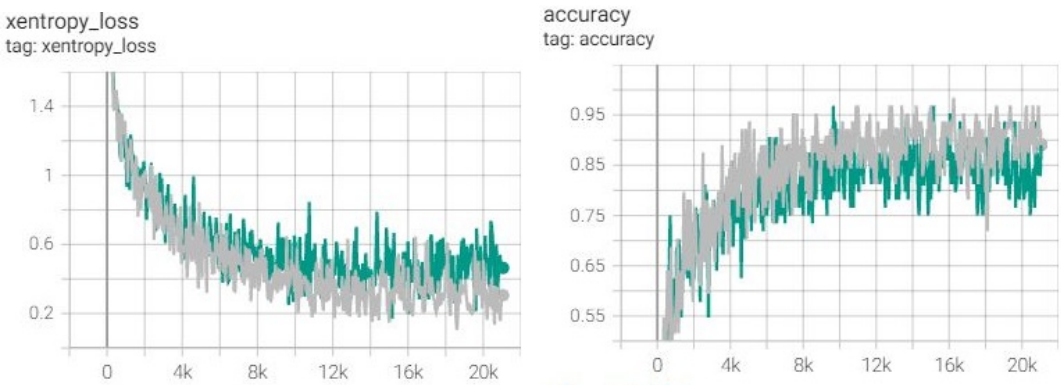
Para esta sección se utilizo la red con 3 capas convolucionales pues fue la que entregó mejores resultados y la probabilidad de Dropout igual a 0.5.
Al comparar los modelos, se puede observar en la figura 4 que en cuanto a los errores de clasificacion, el modelo que presenta el flag data augmentation activado tiene un error de clasificacion considerablemente menor al modelo que tiene le flag data augmentation desactivado (aproximadamente 0.87 vs 1.04 respectivamente) esto indica un posible overfitting del modelo sin el flag activado y en cuanto a los valores de accuracy obtenidos, estos son bastante similares e incluso la mejor Accuracy de validación corresponde al modelo usando este flag activado.
Esto cumple con lo esperado según lo visto en clases, pues el data augmentation nos permite aumentar el tamaño del dataset permitiendo así obtener un mejor entrenamiento, los cuales se pueden observar en la figura 5, y mejores resultados en la validación. Por otro lado en términos de velocidad se obtuvieron resultados similares, pues necesitaron la misma cantidad de iteraciones para el entrenamiento.



(a) Curva de aprendizaje utilizando los datos de validación

(b) Evolución del Accuracy utilizando los datos de validación

Figure 4: Comparación entre los modelos utilizando el flag data Augmentation activado y desactivado. En naranja se muestran los datos del modelo con 3 capas convolucionales. En gris se muestran los datos del modelo con 3 capas convolucionales y el flag data Augmentation activado.



(a) Curva de aprendizaje utilizando los datos de entrenamiento

(b) Evolución del Accuracy utilizando los datos de entrenamiento

Figure 5: Comparación entre los modelos utilizando el flag data Augmentation activado y desactivado, con los datos de entrenamiento. En gris se muestran los datos del modelo con 3 capas convolucionales. En verde se muestran los datos del modelo con 3 capas convolucionales y el flag data Augmentation activado.

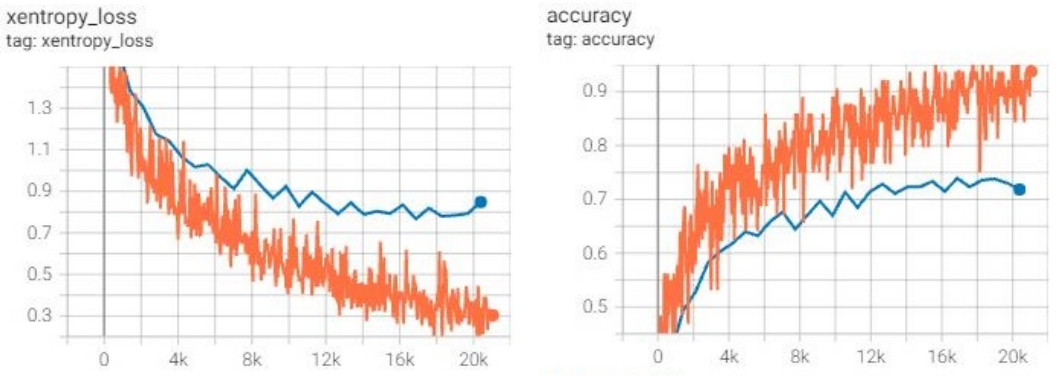
3 Parte de Programación

Para esta sección, se eliminaron las capas fully connected que habían anteriormente, para así de esta manera agregar una capa convolucional con un kernel de 1x1, es decir, una convolución de 1x1 y un número de filtros igual a 10, pues se buscaba generar 10 feature maps (se realizaron los experimentos con la probabilidad de dropout igual a 0.5 y 3 capas convolucionales).

A continuación se presenta un extracto de código que permitió reemplazar las capas fully connected por una sola capa convolucional de 1x1:

```
# Agregamos una capa de convolución:
layers_list.append(
    tf.keras.layers.Conv2D(10,activation = None, kernel_size=(1,1),name='conv1por1',
    bias_initializer=tf.keras.initializers.Constant(value=0.05))
)
layers_list.append(
    tf.keras.layers.AveragePooling2D(pool_size=(4,4),strides=1)
)
layers_list.append(
    tf.keras.layers.Flatten()
)
# Inicializacion de activacion de salida
layers_list.append(
    tf.keras.layers.Activation(tf.nn.softmax))
```

Como se pueden observar en la figura 6, se obtuvo al rededor de un 0.7309 como el mejor accuracy con los datos de validación, lo cual comparando con los resultados anteriores correspondientes a 4, son bastante similares a los obtenidos con las 3 capas convolucionales + las capas fully connected (esto con respecto a los resultados del mejor valor de Accuracy en validacón). Si comparamos ahora las curvas de aprendizaje, esta es comparable con la red con 3 capas convolucionales + capas fully connected (incluso se llegan a mejores valores de tasa de accuracy en validación), sus curvas son bastante similares, pero se puede observar menores valores de la tasa de error en este ultimo experimento (sin capas fully connected), demostrando que es posible realizar la clasificación sin capas fully connected y además se puede concluir que se llega a una mejor clasificación utilizando el modelo que sustituye las capas fully connected por la capa convolucional de 1x1.



(a) Curva de aprendizaje

(b) Evolución del Accuracy

Figure 6: En naranja se muestran los valores de entrenamiento y en azul los valores de validación.

3.1 Comparación de espacio en memoria

En la figura 1 se puede observar que comparten los parámetros de las 3 capas convolucionales correspondientes a 7.079.424, luego en vez de sumarle los parámetros de las capas fully connected, se le suman los parámetros de la ultima capa convolucional de 1x1, a partir de este cálculo se puede obtener que como era esperado visto en clases, el modelo que tiene solo capas convolucionales posee menos parámetros y a su vez necesita menos memoria para guardar estos. Como vimos en clases y como se puede observar en la parte teórica, la mayoría de los parámetros (pesos + bias) de la red convolucional que tiene 3 capas fully connected se encuentran en las capas fully connected.

	Numero de parámetros	Espacio en memoria de los parámetros
convnet con capas fully-connected	138357544	553.43 Megabytes
convnet solo con capas convolucionales	7079424+ (1x1x512x10+10)	0.8856 Megabytes

Table 1: Tabla comparativa número de parámetros y memoria

4 Anexos

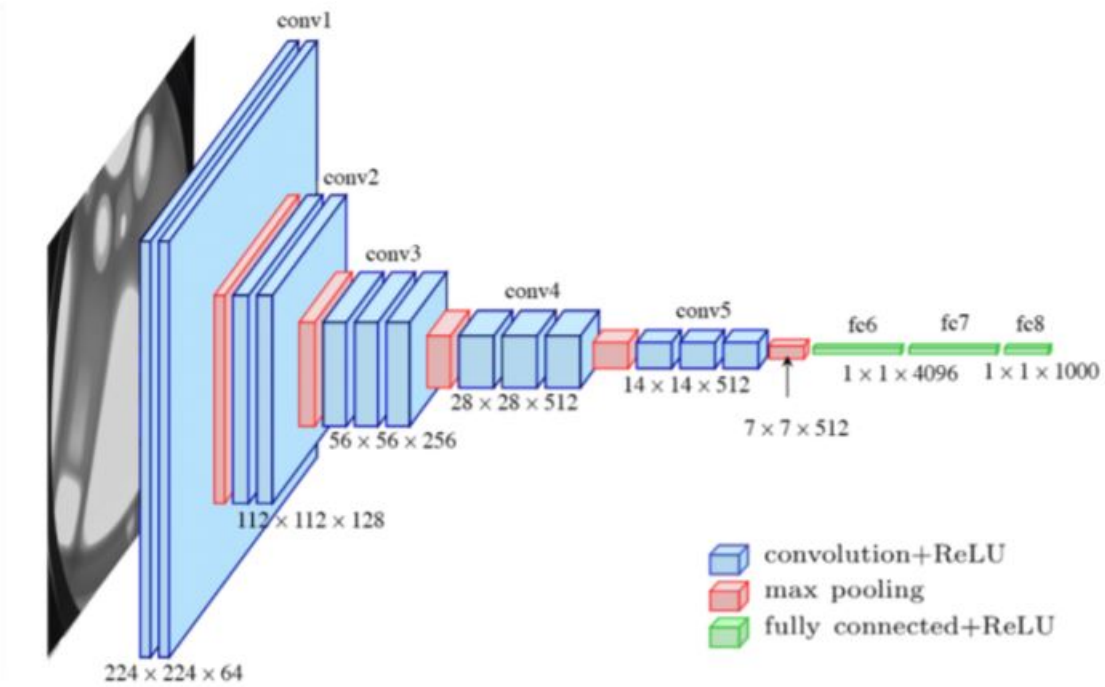


Figure 7: Diseño red vgg-16 la cual fue analizada en la parte teórica