Profesor: Dr. Oldemar Rodríguez Rojas
Análisis de Datos 2
Fecha de Entrega: Domingo 27 de noviembre a las 12 media noche
Instrucciones:

- Las tareas son estrictamente individuales.
- Tareas idénticas se les asignará cero puntos.
- Todas las tareas tienen el mismo valor en la nota final del curso.
- Cada día de entrega tardía tendrá un rebajo de 20 puntos.

Tarea Número 13

1. [10 puntos] Basado en el código visto en el documento de la clase sobre las convoluciones, investigue y aplique al menos 3 núcleos o filtros diferentes para la imagen `einstein.csv`. Muestre las imágenes generadas.

2. [15 puntos] La siguiente notación significa lo siguiente:

   a) CONV-K-N denota una capa convolucional con N Filtros, cada uno de tamaño. $K \times K$, los parámetros "padding" y "stride" serán siempre 0 y 1 respectivamente.

   b) POOL-K indica una capa de "pooling" de tamaño $K \times K$ con "stride" K y "padding" 0.

   c) FLATTEN denota la capa aplanada.

   d) FC-N representa una capa completamente conectada (Fully-Connected) con N neuronas.

   En la siguiente tabla, para cada capa, se muestra la cantidad de variables generadas ("feature mapping"), es decir, el tamaño de las matrices. También se muestra el número de parámetros entrenables en cada capa.

Capa	Dimensión del "feature mapping"	Número de parámetros entrenables
INPUT	$128 \times 128 \times 1$	0
CONV-9-32	$120 \times 120 \times 32$	2624
POOL-2	$60 \times 60 \times 32$	0
CONV-5-64	$56 \times 56 \times 64$	1664
POOL-2	$28 \times 28 \times 64$	0
CONV-5-64	$24 \times 24 \times 64$	1664
POOL-2	$12 \times 12 \times 64$	0
FLATTEN	9216	0
FC-3	3	27648

   Para la tabla anterior explique cómo se obtuvo cada uno de los valores numéricos que aparecen en cada casilla.

3. [15 puntos] Para cada capa, calcule la cantidad de variables generadas ("feature mapping"), es decir, el tamaño de las matrices. Además, calcule el número de parámetros entrenables. La notación sigue la siguiente convención:

a) CONV-K-N denota una capa convolucional con N Filtros, cada uno de tamaño. $K \times K$, los parámetros "padding" y "stride" serán siempre 0 y 1 respectivamente.

b) POOL-K-S indica una capa de "pooling" de tamaño $K \times K$ con "stride" (zancada) S y "padding" 0.

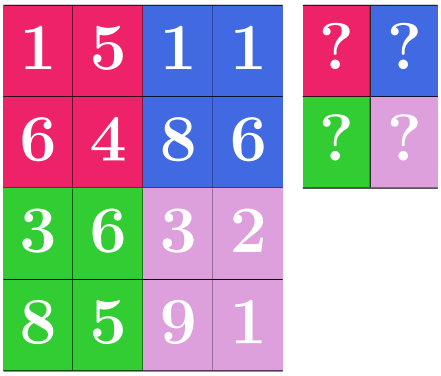
c) ReLU indica la aplicación de la función de activación ReLU.

d) FLATTEN denota la capa aplanada.

e) FC-N representa una capa completamente conectada (Fully-Connected) con N neuronas.

Capa	Dimensión del "feature mapping"	Número de parámetros entrenables
INPUT	$32 \times 32 \times 3$	0
CONV3-8		
ReLU		
POOL-2-2		
CONV3-16		
ReLU		
POOL-3-1		
FLATTEN		
FC-10		

4. [10 puntos] Complete en el siguiente gráfico la matriz de la derecha, para esto utilice mean-pooling, es decir, agrupación de capas mediante el promedio:



5. [25 puntos] Descomprima el archivo `piedrapapeltijera.zip`. Este conjunto de imágenes tiene 3 clases: `piedra, papel` y `tijera`. El objetivo del conjunto de datos es identificar la forma de la mano. Con la tabla de datos realice lo siguiente:

a) Cargue las imágenes contenidas en el archivo `piedrapapeltijera.zip` y obtenga la etiqueta correspondiente para cada imagen.

b) Divida la tabla utilizando un 90 % para entrenamiento y un 10 % para pruebas. Luego normalice las tablas y aplique el `one-hot-encoding` a las etiquetas.

c) Imprima una imágen de cada clase de la tabla de entrenamiento.

d) Genere la estructura de Redes Neuronales Convolucionales. Para esto utilice al menos 2 capas de convolución `2D` y 2 capas de `Maxpooling`, luego utilice `flatten()`, `Dropout()` y al menos 2 capas ocultas. Muestre un resumen del modelo.

e) Configure el modelo con la función de costo `categorical_crossentropy`, la función de optimización `adam` y la métrica `accuracy`.

f) Entrene el modelo con los siguientes parámetros `epochs=50` y `batch_size=64`. Esto puede tardar un rato.

g) Utilice el método `evaluate()` para evaluar las predicciones y el error global. Establezca el valor de 0 para el parámetro `verbose` para omitir la salida en consola.

h) Obtenga y cargue 10 fotos de su mano haciendo formas de `piedra, papel` y `tijera`. Luego con ayuda del modelo realice una predicción de esas 10 imágenes. Muestre y comente los resultados.

i) Repita nuevamente los pasos anteriores, pero agregue a las capas de convolución el siguiente parámetro: `kernel_initializer = 'glorot_uniform'`. ¿Mejoran los resultados? Explique e investigue sobre el parámetro `kernel_initializer`.

6. [25 puntos] Vamos a usar las tablas de datos `ZipDataTestCod.csv` y `ZipDataTrainCod.csv`, la primera columna guarda la información de tipo de dígito es: un número del 0 al 9, mientras que las otras 256 columnas guardan la información de cada pixel, para una imagen de 16x16 de un solo canal. Para esto realice lo siguiente:

   a) Cargue las tablas de datos `ZipDataTestCod.csv` `ZipDataTrainCod.csv` en `Python`.

   b) Genere las tablas de las categorías (`Y_train` y `Y_test`) y paselas a la forma **one-hot-encoding**

   c) Genere las tablas de las imágenes (`X_train` y `X_test`) realizando un `reshape(-1, 16,16,1)` con `numpy` y seguidamente normalice dichas tablas.

   d) Imprima una imagen de la tabla `X_train`.

   e) Genere un modelo de Redes Neuronales Convolucionales. Para esto utilice 2 capas de convolución 2d y 2 capas de Maxpooling, luego utilice `flatten()`, `Dropout()` y al menos 2 capas ocultas. Utilice la función de optimización `RMSprop`, la función de costo `categorical_crossentropy` y la métrica `accuracy`.

   f) Haga un resumen del modelo.

   g) Entrene el modelo con los siguientes parámetros `epochs=60` y `batch_size=18`. Esto puede tardar un rato.

   h) Utilice el método `evaluate()` para evaluar las predicciones y el error global. Establezca el valor de 0 para el parámetro `verbose` para omitir la salida en consola.

   i) Obtenga las predicciones sobre la tabla de testing utilizando la función `predict()`. Genere la matriz de confusión, obtenga la precisión global y por categoría. Muestre y comente los resultados.

   j) Tome 4 imágenes de la tabla de testing, muéstrelas y realice una predicción con el modelo generado.


oldemar rodríguez
CONSULTOR en M1NΣR14 DΣ D4TØS