Proyecto 2
Reconocimiento de comandos de voz mediante espectrogramas y aprendizaje profundo

Fecha de asignación:	15 de noviembre, 2022	Fecha de entrega:	6 de diciembre, 2022
Grupos:	2 ó 3 personas	Enlace a la asignación:	GitHub Classroom

1. Objetivo general

Al finalizar el proyecto, el estudiande habrá combinado técnicas clásicas de procesamiento digital de señales con técnicas modernas de aprendizaje profundo para prototipar un sistema sencillo de reconocimiento de comandos de voz.

2. Objetivos específicos

1. Construir el espectrograma de muestras de audio para obtener una representación 2D de las mismas.

2. Aplicar un Banco de Filtros de Frecuencias de Mel para emular la percepción del oído humano.

3. Entrenar una Red Neuronal Convolucional para el reconocimiento de comandos de voz.

4. Evaluar el desempeño de la Red Neuronal Convolucional ante nuevas entradas.

3. Descripción General

Los avances en la capacidad computacional reciente han abierto paso a que técnicas modernas de aprendizaje de máquina, en especial aprendizaje profundo (deep learning), jueguen un rol protagonista en la resolución de problemas complejos en el área de procesamiento digital de señales e imágenes. Sin embargo, debe entenderse las mismas como un complemento a las técnicas de procesamiento clásico estudiadas en este curso, y no como un sustituto. El presente proyecto tiene como objetivo que el estudiante experimente una breve pincelada del tema de aprendizaje profundo orientado a la resolución de un problema típico en el procesamiento digital de señales.

Se desea desarrollar un sistema capaz de reconocer comandos de voz en grabaciones de audio. Para ello se hará uso de una base de datos preexistente que facilite el entrenamiento y la evaluación del sistema. La base de datos consta de múltiples grabaciones de diferentes comandos en

Cuadro 1: Comandos a reconocer en el proyecto

Clase	Comando
0	*yes*
1	*no*
2	*up*
3	*down*
4	*left*
5	*right*
6	*on*
7	*off*
8	*stop*
9	*go*
10	desconocido
11	silencio/ruido de fondo

inglés. Para este proyecto se identificarán las muestras de audio en una de las siguientes clases mostradas en la Tabla 1.

La categoría *desconocido* corresponde a muestras con palabras que no sean reconocidas como ninguno de los comandos y *silencio/ruido* de fondo corresponde a muestras silenciosas o con ruido de fondo donde no exista una palabra fácilmente perceptible.

Para facilitar el problema cada muestra de audio tiene una duración fija de 1 segundo con la palabra a clasificar. La Tabla 2 resume las características de las señales de audio. Es posible que algunas de las muestras no cumplan con la duración propuesta, en cuyo caso se deberá ajustar rellenando con ceros, por ejemplo.

Cuadro 2: Características de las muestras de audio

Propiedad	Valor
Contenedor de audio	WAV
Duración	1s
Endianness	LE
Ancho de muestra	16bits
Frecuencia de muestreo	16kHz
Canales	1

Cada muestra debe ser convertida en un espectrograma con lo que se tendrá una representación bidimensional de la señal de audio. Con el fin de emular la percepción del sistema auditivo humano, dicho espectro deberá ser escalado utilizando un Banco de Filtros de Frecuencias de Mel. La Figura 1 muestra dicho concepto.

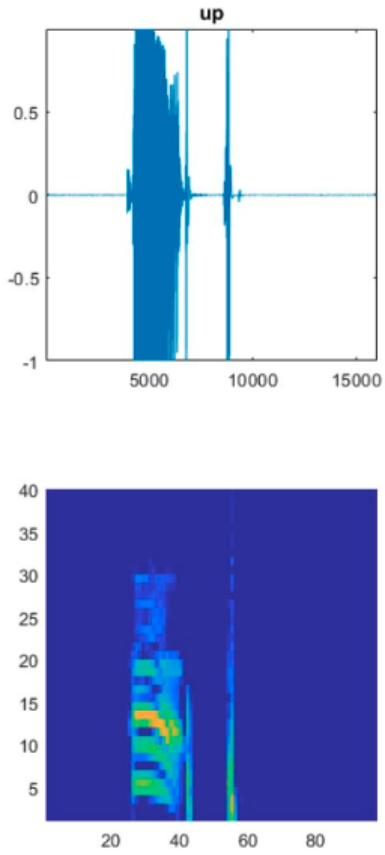

Figura 1: Señal de audio original (arriba) y su espectrograma (abajo).

Con el fin de encontrar patrones en dichos espectrogramas que permitan reconocer comandos de voz, se entrenará una Red Neuronal Convolucional (CNN). La red recibirá como entrada el espectrograma y tendrá como salida las 12 clases de la Tabla 1. Cada clase posee un valor numérico que representa la probabilidad de que la señal de voz corresponda a cierto comando. La Figura 22 muestra el concepto anterior.

En este ejmplo, con el espectrograma dado la red neuronal predice que existe un 84,59 % de probabilidad de que se trate del comando *up*.

El presente proyecto está basado en el ejemplo de Reconocimiento de Comandos Voz Utilizando Aprendizaje Profundo de Matlab. Puede referirse al mismo para obtener una guía más detallada.

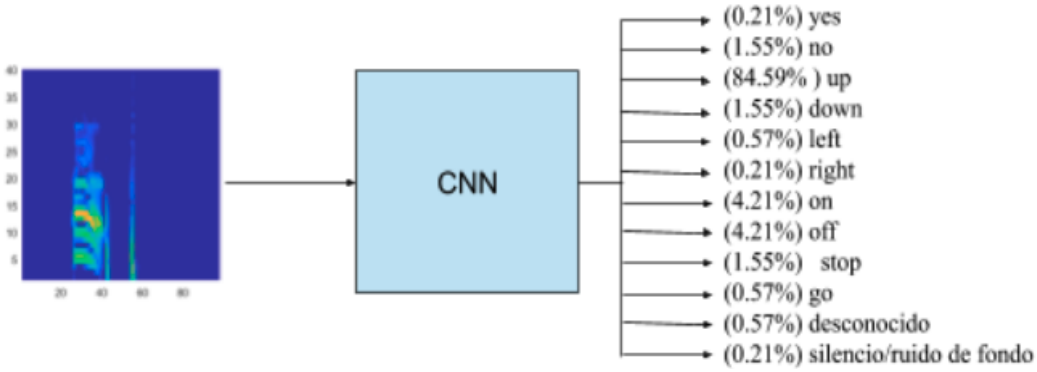

Figura 2: Diagrama general de la solución propuesta

4. Descripción de Módulos

4.1. Espectrograma

La Tabla 3 presenta una posible configuración (que ha sido probada) para generar el espectrograma de cada clip de audio. Es posible variar dichos valores, sin embargo tenga en cuenta que las etapas posteriores podrían estar ajustadas a los mismo y, por tanto, podrían requerir variaciones también.

Cuadro 3: Valores a utilizar para generar el espectrograma

Propiedad	Valor
Duración de cada clip	1s
Duración de cada columna	25ms
Distancia entre columnas	10ms
Bandas de Mel	40

Si bien la duración de cada columna es de $25ms$, note que la distancia de $10ms$ entre cada una implica un traslape temporal entre columnas. Finalmente, el número de bandas de Mel corresponderá al largo de las columnas del espectrograma. Refiérase de nuevo a la Figura1 para un ejemplo de un espectrograma generado con estas características.

Finalmente, con el fin de obtener una distribución más suave, considere aplicar tomar el logaritmo del espectrograma

$$spec_{10} = \log_{10}(spec + \epsilon)$$

donde ϵ es un desplazamiento arbitrario pequeño.

Implemente una arquitectura modular, cuyos bloques funcionales cumplan con los requerimientos citados a continuación.

4.2. Red Neuronal Convolucional

Se recomienda la arquitectura en la Tabla 4 para la red neuronal convolucional:

Cuadro 4: Arquitectura de la Red Neuronal propuesta

#	Capa	Cantidad de Filtros	Tamaño de Filtro	Padding	Stride	Dropout
1	Input					
2	Convolution2D	12	3x3	*Same*		
3	Batch Normalization					
4	ReLU					
5	Max Pooling 2D		3x3	*Same*	2x2	
6	Convolution 2D	24	3x3	*Same*		
7	Batch Normalization					
8	ReLU					
9	Max Pooling 2D		3x3	*Same*	2x2	
10	Convolution 2D	48	3x3	*Same*		
11	Batch Normalization					
12	ReLU					
13	Max Pooling 2D		3x3	*Same*	2x2	
14	Convolution 2D	48	3x3	*Same*		
15	Batch Normalization					
16	ReLU					
17	Convolution 2D	48	3x3	*Same*		
18	Batch Normalization					
19	ReLU					
20	Max Pooling 2D		1x13		1x1	
21	Dropout					0.2
22	Dense	12				
23	Softmax					

Tenga en cuenta que cada biblioteca implementa las capas con distinto nombre y con diferentes propiedades. También considere que ciertas bibliotecas pueden tomar capas como ReLU o Softmax como funciones de activación en lugar de capas independientes.

Se recomienda utilizar los parámetros de la Tabla 5 para configurar el entrenamiento de la red neuronal.

5. Procedimiento

1. Descargue el set de entrenamiento y validación del sitio de TensorFlow.

Cuadro 5: Parámetros recomendados para el entrenamiento

Propiedad	Valor
Optimizador	Adam
Función de pérdida	Cross Entropy
Tasa de aprendizaje	3e-4
Épocas	25
Tamaño de bache	128
Barajado de épocas	Sí

2. Agrupe las muestras de audio en las diferentes clases dadas en la Tabla 1.

   a) Aquellas muestras que no correspondan a ningún comando deberán clasificarse como "desconocido".

   b) Considere que tendrá muchas más muestras de la clase "desconocido". Procure cargar una cantidad de muestras de dicha clase tal que la cantidad sea similar a las demás clases.

3. Separe las muestras en dos conjuntos:

   a) Conjuntos de validación: muestras listadas en el archivo *validation_list.txt*.

   b) Conjunto de entrenamiento: el resto.

   c) Como su nombre lo indica, utilice el conjunto de entrenamiento únicamente para entrenar, y el de validación para caracterizar el desempeño de su red. Nunca mezcle conjuntos.

4. Calcule los espectrogramas de todas las muestras:

   a) Utilice la descripción y los parámetros de la Sección 4.1.

   b) Este puede ser un proceso costoso y tardado, por lo que se recomienda guardar dichos espectrogramas en archivos para poder ser utilizados posteriormente sin necesidad de calcularlos nuevamente.

5. Valide visualmente los datos preprocesados:

   a) Grafique la señal de audio de diferentes comandos.

   b) Grafique sus respectivos espectrogramas.

   c) Grafique el histograma acumulativo de valores de espectrograma de las muestras.

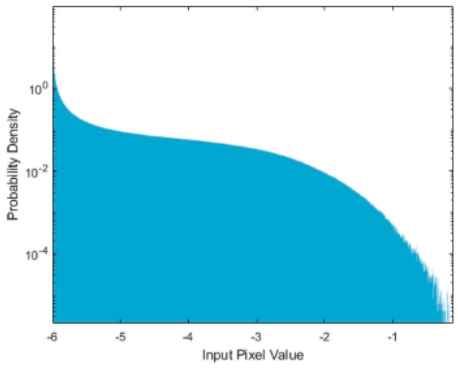

Figura 3: Ejemplo de histograma de valores en el espectrograma

d) Se espera que el histograma sea suave. En caso contrario, asegúrese que la cantidad de muestras de cada clase sea equitativa (paso 2). La Figura 3 presenta un ejemplo de dicho histograma acumulativo (note las escalas).

6. Genere muestras de ruido de fondo:

   a) Tome segmentos de 1s de largo a partir de los archivos de audio en la carpeta *_background_noise_*.

   b) Procure que el número de muestras sea congruente con la distribución del resto de clases.

   c) Tome igual cantidad de muestras de cada archivo en dicha carpeta.

   d) Si lo desea puede agregar sus propias muestras de ruido de fondo.

   e) Puede utilizar técnicas de aumento de datos para tomar estas muestras de manera más ágil (por ejemplo replicar las muestras a diferentes volúmenes).

   f) Distribuya estas muestras generadas entre el set de entrenamiento y validación.

7. Grafique la distribución de las muestras del conjunto de entrenamiento y validación.

   a) La Figure 4 muestra un ejemplo:

8. Opcional: Aumente sus conjuntos de entrenamiento y validación utilizando técnicas de aumento de datos.

9. Construya la red neuronal utilizando la descripción en la Sección 4.2.

   a) Puede utilizar cualquier biblioteca que desee (mientras sea de acceso y código libre).

   b) Por su simplicidad, documentación y cantidad de ejemplos disponibles se recomienda utilizar la biblioteca Keras o PyTorch de Python.

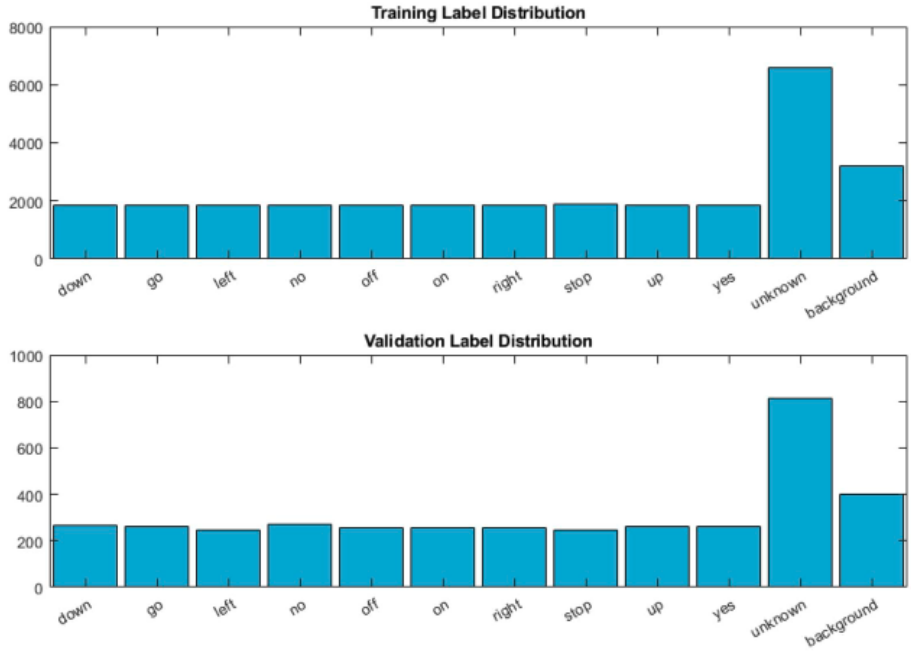

Figura 4: Ejemplo de distribución final de muestras

c) Si bien este no es un curso de reconocimiento de patrones, procure hacerse una intuición de la función de las diferentes capas en la red.

.

10. Entrene la red construida utilizando su conjunto de entrenamiento

    a) Dependiendo sus capacidades computacionales, este proceso puede demorar varios minutos.

    b) Procure guardar el resultado de su entrenamiento (modelo) para ser utilizado luego.

    c) Grafique el historial de la evolución de la métrica de exactitud (*accuracy*) tanto de los datos de entrenamiento como de validación.

    d) Grafique el historial de la evolución de la métrica de pérdida (*loss*) tanto de los datos de entrenamiento como de validación.

    e) La Figura 5 muestra un ejemplo de dichas gráficas.

11. Evalúe la red tanto en el conjunto de entrenamiento como en el de validación.

    a) Presente el resultado de la evaluación con alguna métrica de error (como error cuadrático medio, por ejemplo).

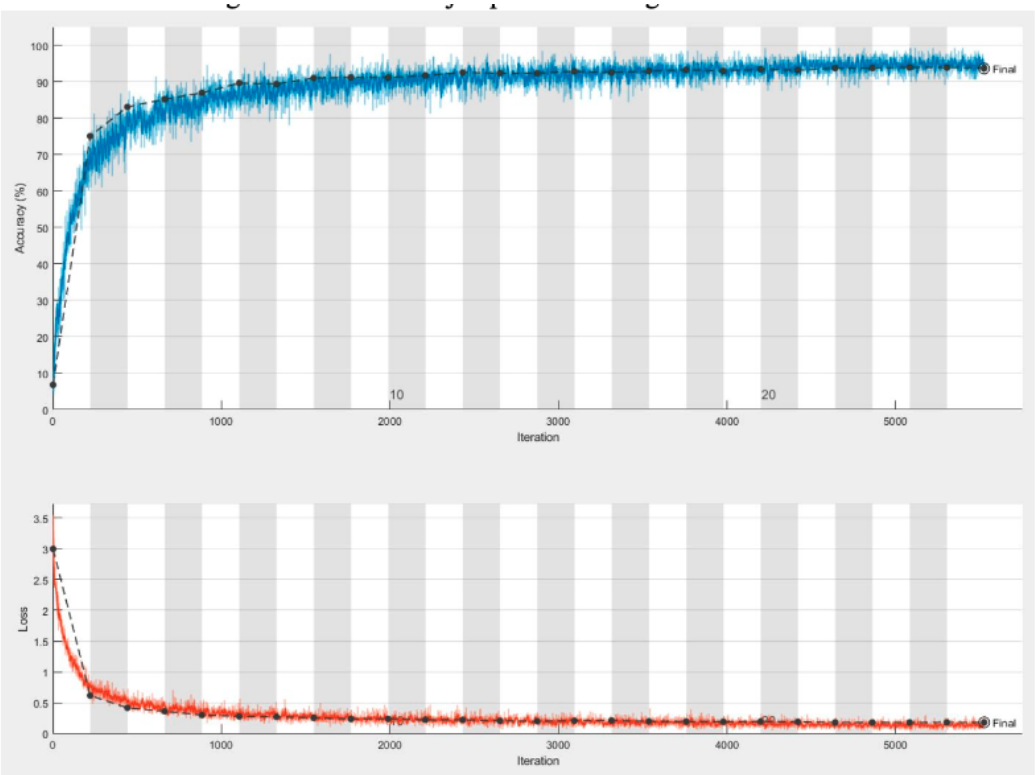

Figura 5: Ejemplo de la evolución de la precisión y la pérdida

12. Analice el desempeño de su red neuronal

    a) Presente la matriz de confusión.

    b) Presente valores de *precision* y *recall* mediante resúmenes de filas y columnas.

    c) Ordene las clases en su matriz de confusión.

    d) Obtenga conclusiones relevantes de esta matriz.

    e) La Figura 6 muestra un ejemplo de dicha matriz.

13. Tome mediciones de desempeño de su red:

    a) Cuántos espectrogramas por segundo puede procesar.

    b) Cuál es el consumo de CPU.

    c) Si está usando un GPU, cuál es el consumo de GPU.

14. Opcional: Si desea mejorar la precisión o el desempeño de su red, puede regresar al paso 9 y modificar la red:

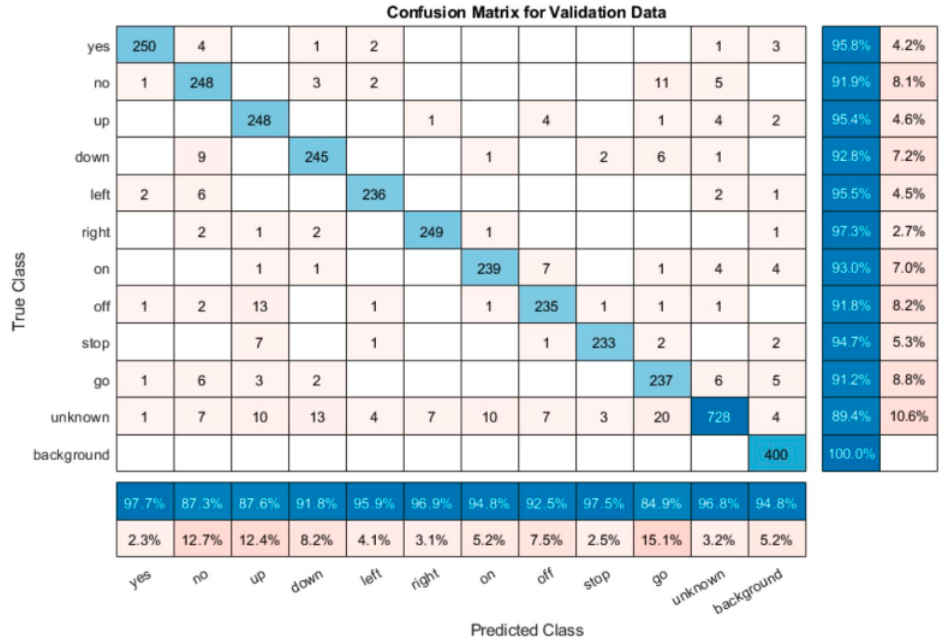

Figura 6: Ejemplo de matriz de confusión

a) Puede agregar o quitar capas.

b) Puede modificar los hyperparámetros (tamaño de filtros, número de filtros, stride, etc..)

c) Puede modificar los parámetros de entrenamiento.

15. Con el fin de evaluar la capacidad de generalización de su red, obtenga métricas de desempeño con audios grabados por usted.

6. Entregables

1. Informe en formato .PDF y con estructura de artículo científico IEEE Transactions de no más de 7 páginas que incluya:

   a) Introducción y estado del arte.

   b) Fundamentos teóricos

      1) Transformada de Fourier de Corto Plazo
      2) Espectrogramas
      3) Escalas de Mel
      4) Conceptos básicos sobre aprendizaje profundo

5) Redes Neuronales Convolucionales

c) Descripción del sistema.

d) Resultados y análisis (basado en los pasos de la Sección 5).

e) Conclusiones y recomendaciones.

2. Código fuente documentado necesario para la ejecución del proyecto, así como referencias a bibliotecas externas utilizadas, de ser el caso.

3. Demostración del sistema funcionando, mediante un video breve y conciso.

Las secciones del artículo del científico deben concidir con aquellas del formato IEEE Transactions y no con los puntos mencionados anteriormente. Éstos son únicamente ideas que deben estar detalladas en la sección que se considere pertinente.