PROYECTO DL

Fredy Velasquez
Angel Higueros

DESCRIPCIÓN DEL PROBLEMA

El acceso al contenido multimedia es fundamental, pero la **comunidad con discapacidad auditiva aún enfrenta barreras**.

Aunque el lenguaje de señas ha sido esencial, su **implementación depende** de **intérpretes humanos**, limitando la autonomía.

Abordar esto es clave para **mejorar la accesibilidad** y la **disponibilidad de contenido inclusivo.**

PROPUESTA DE SOLUCIÓN

Desarrollo de un modelo de aprendizaje profundo especializado en procesar videos de habla clara para traducir movimientos labiales a texto con alta precisión en entornos controlados.

ASPECTOS A TOMAR EN CONSIDERACIÓN

CONJUNTO DE DATOS

Conjunto de **datos GRID** cuenta con **5 secciones** dedicadas al **estudio de la fonética** categorizadas y agregadas a un espacio en google drive

USO DE GOOGLE COLAB PRO

Se pagó $10 para contar con más potencia computacional y recursos en general. Al pagar dicha cantidad se pudo utilizar una GPU A100 de NVIDIA que disminuyó el tiempo de entrenamiento considerablemente.

HERRAMIENTAS APLICADAS

vistas en clase

1. TensorFlow

2. Numpy

3. Matplotlib

HERRAMIENTAS APLICADAS

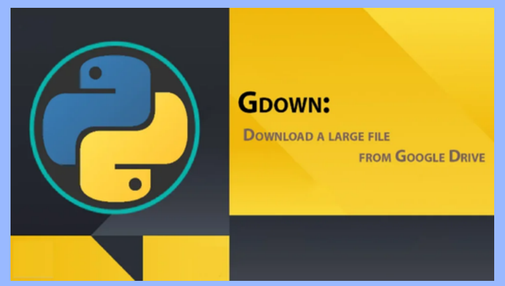
no vistas en clase

1 CV2

2 OpenCV

3 Imageio

1 Gdown



2 Typing

FASES DEL MODELO

01

Importación

Se importan todas las librerías y módulos que serán necesarios para el proyecto.

02

Configuración

Se configura que se usará GPU como dispositivo físico para la ejecución del programa

03

Construir las funciones para cargar los datos

Se crean dos funciones, una para cargar los videos y una para preprocesar las anotaciones

04

Construir el pipeline de datos

Conjunto de datos en TensorFlow es creado y examinado utilizando un iterador y la función “next”

05

Diseño de la deep neural network

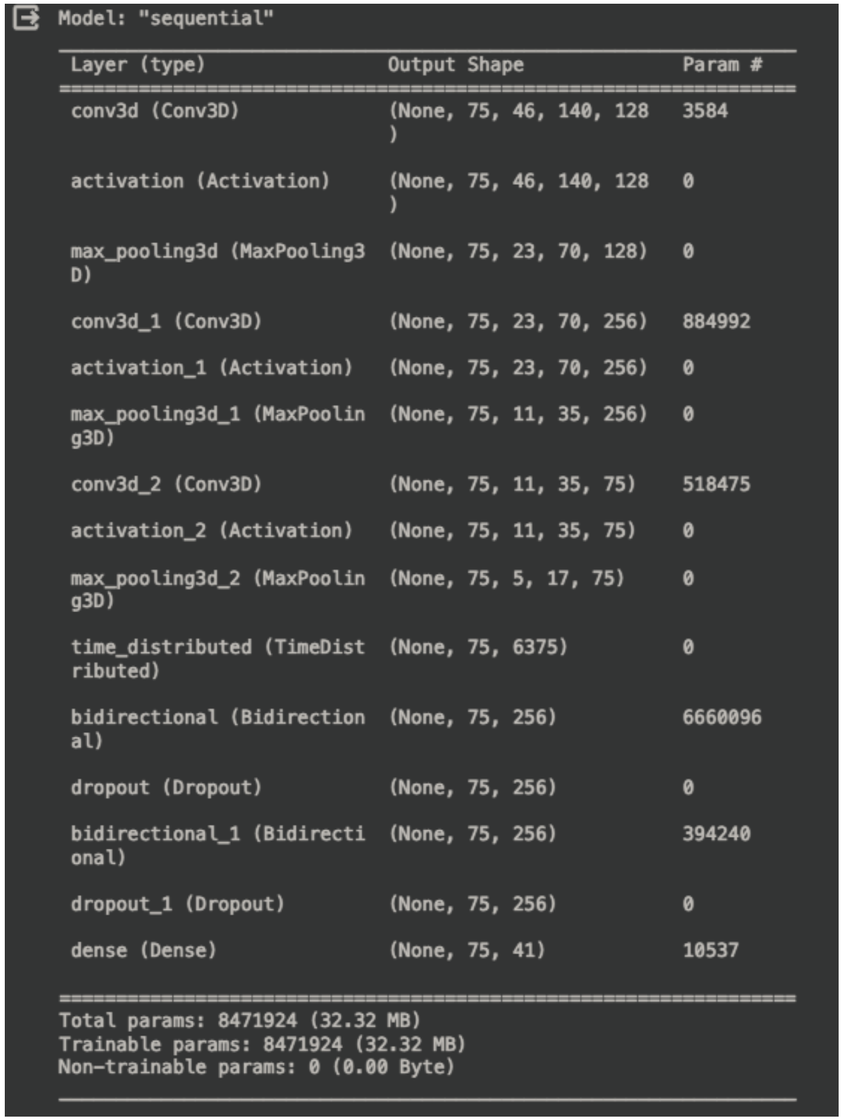
Implementación de un modelo de red neuronal utilizando Keras

06

Configuración y ejecución del entrenamiento

Se importan todas las librerías y módulos que serán necesarios para el proyecto.

FIGURA 1. RESUMEN DEL MODELO UTILIZADO.



```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv3d (Conv3D)             (None, 75, 46, 140, 128   3584
                             )

 activation (Activation)     (None, 75, 46, 140, 128   0
                             )

 max_pooling3d (MaxPooling3  (None, 75, 23, 70, 128)   0
 D)

 conv3d_1 (Conv3D)           (None, 75, 23, 70, 256)   884992

 activation_1 (Activation)   (None, 75, 23, 70, 256)   0

 max_pooling3d_1 (MaxPoolin  (None, 75, 11, 35, 256)   0
 g3D)

 conv3d_2 (Conv3D)           (None, 75, 11, 35, 75)    518475

 activation_2 (Activation)   (None, 75, 11, 35, 75)    0

 max_pooling3d_2 (MaxPoolin  (None, 75, 5, 17, 75)     0
 g3D)

 time_distributed (TimeDist  (None, 75, 6375)          0
 ributed)

 bidirectional (Bidirection  (None, 75, 256)           6660096
 al)

 dropout (Dropout)           (None, 75, 256)           0

 bidirectional_1 (Bidirecti  (None, 75, 256)           394240
 onal)

 dropout_1 (Dropout)         (None, 75, 256)           0

 dense (Dense)               (None, 75, 41)            10537

=================================================================
Total params: 8471924 (32.32 MB)
Trainable params: 8471924 (32.32 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

07

Hacer predicciones

Se lleva a cabo la tarea de hacer una predicción con un modelo de aprendizaje profundo.

8

Tests con videos

Se realiza una prueba de un modelo de aprendizaje profundo en un video

RESULTADOS DEL ENTRENAMIENTO

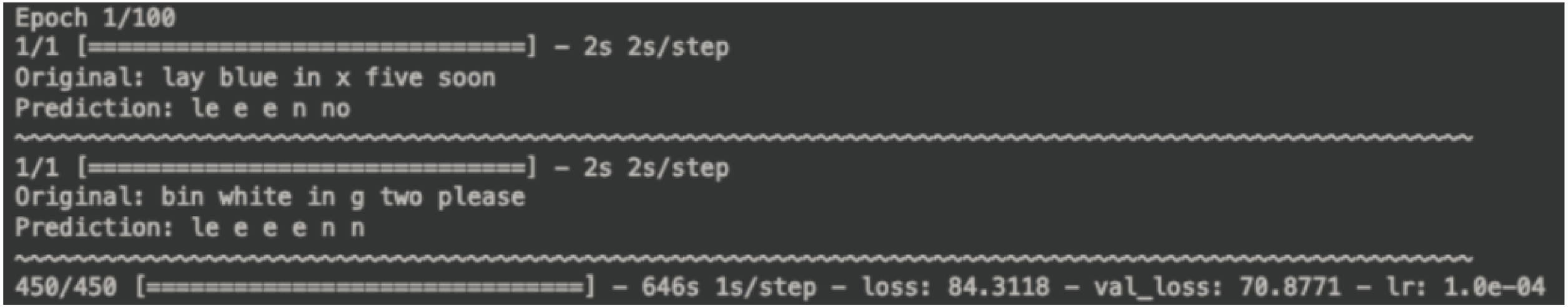


Figura 5. Resultados de la época número 1 del entrenamiento.

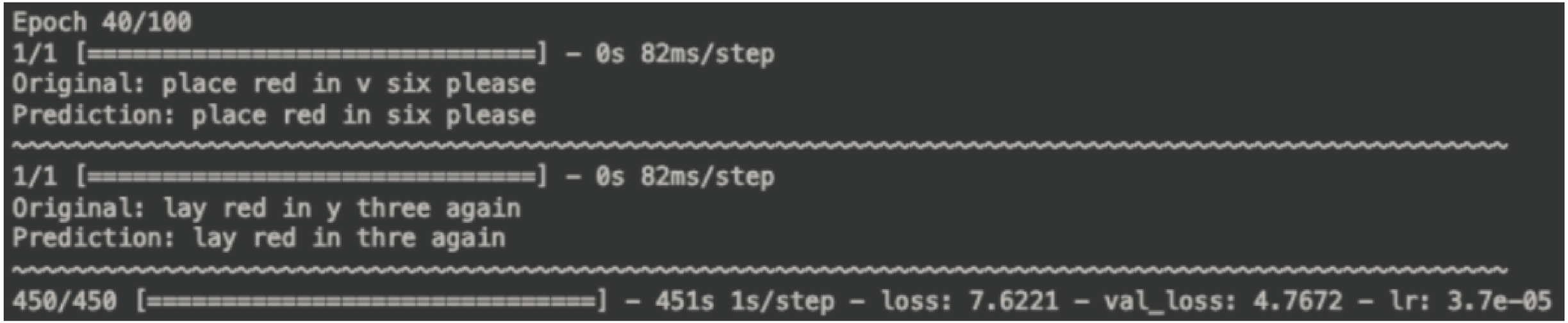


Figura 6. Resultados de la época número 40 del entrenamiento.

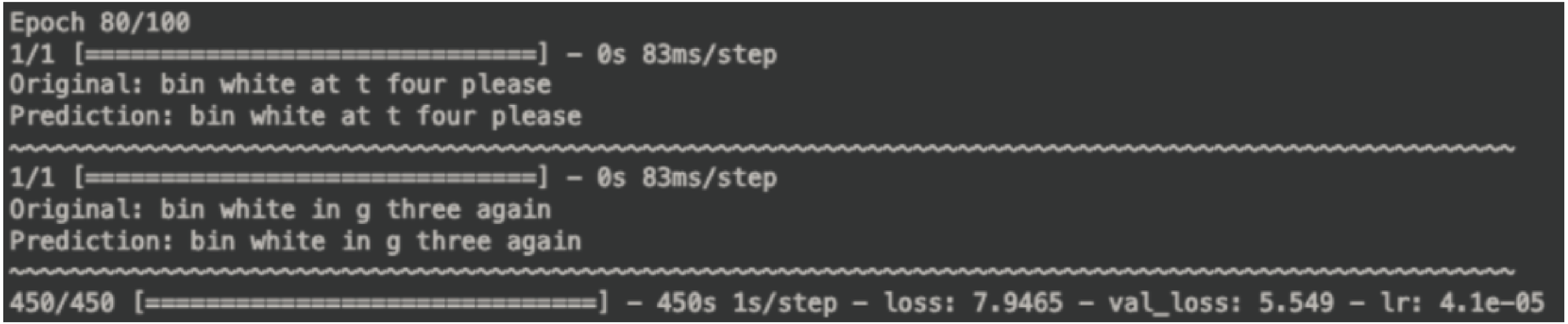
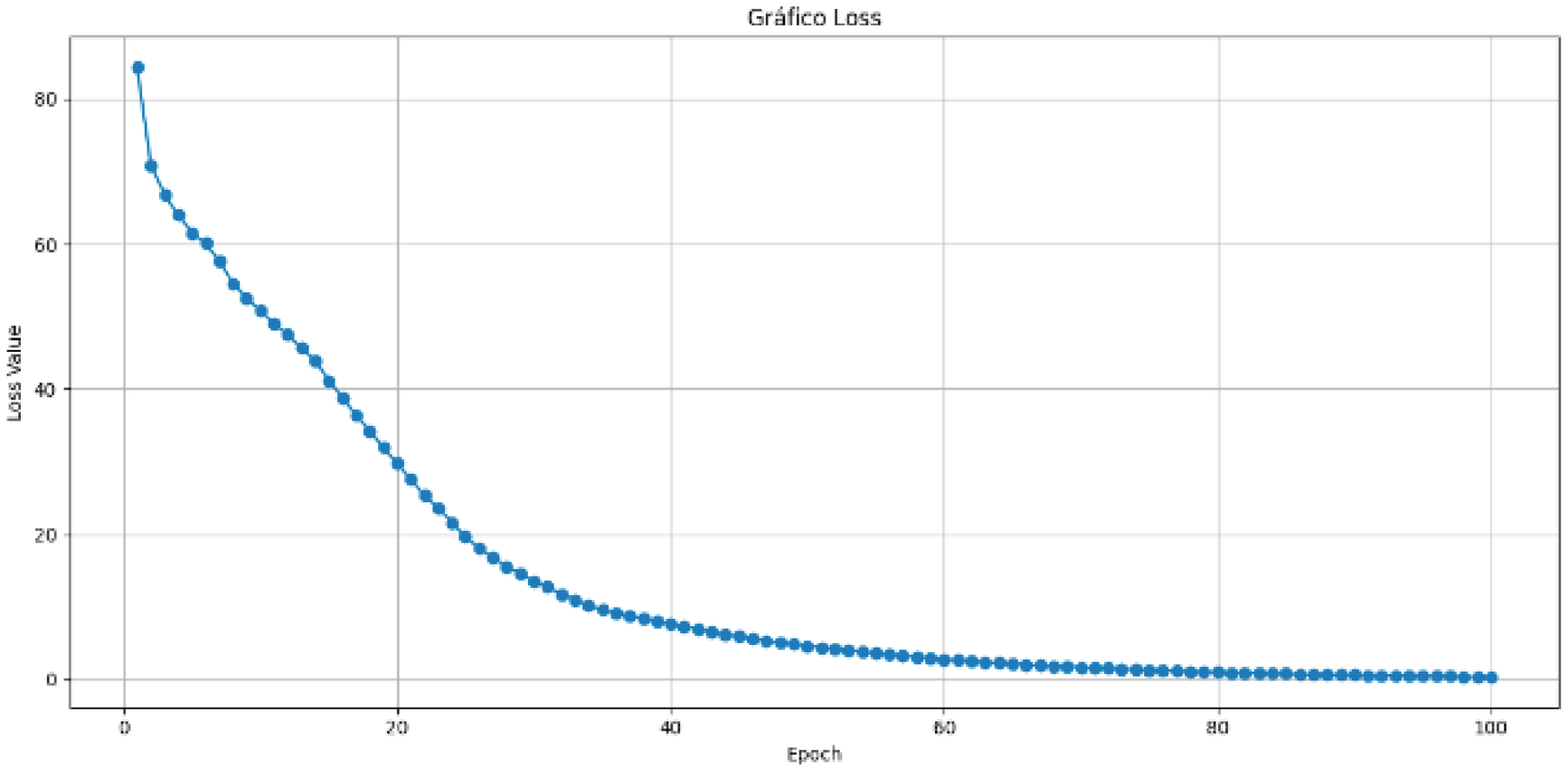


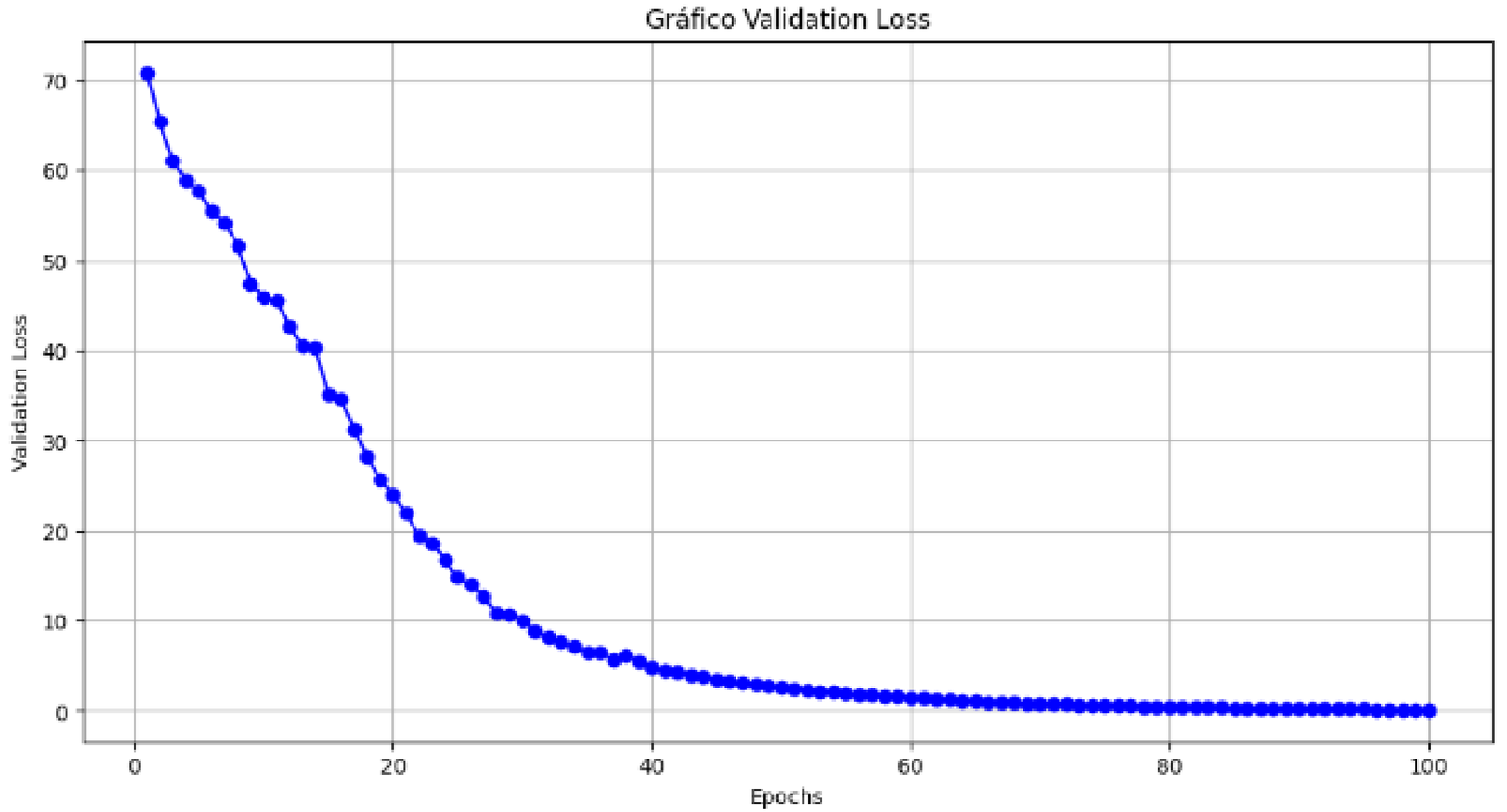
Figura 7. Resultados de la época número 80 del entrenamiento.

RESULTADOS (MÉTRICAS)

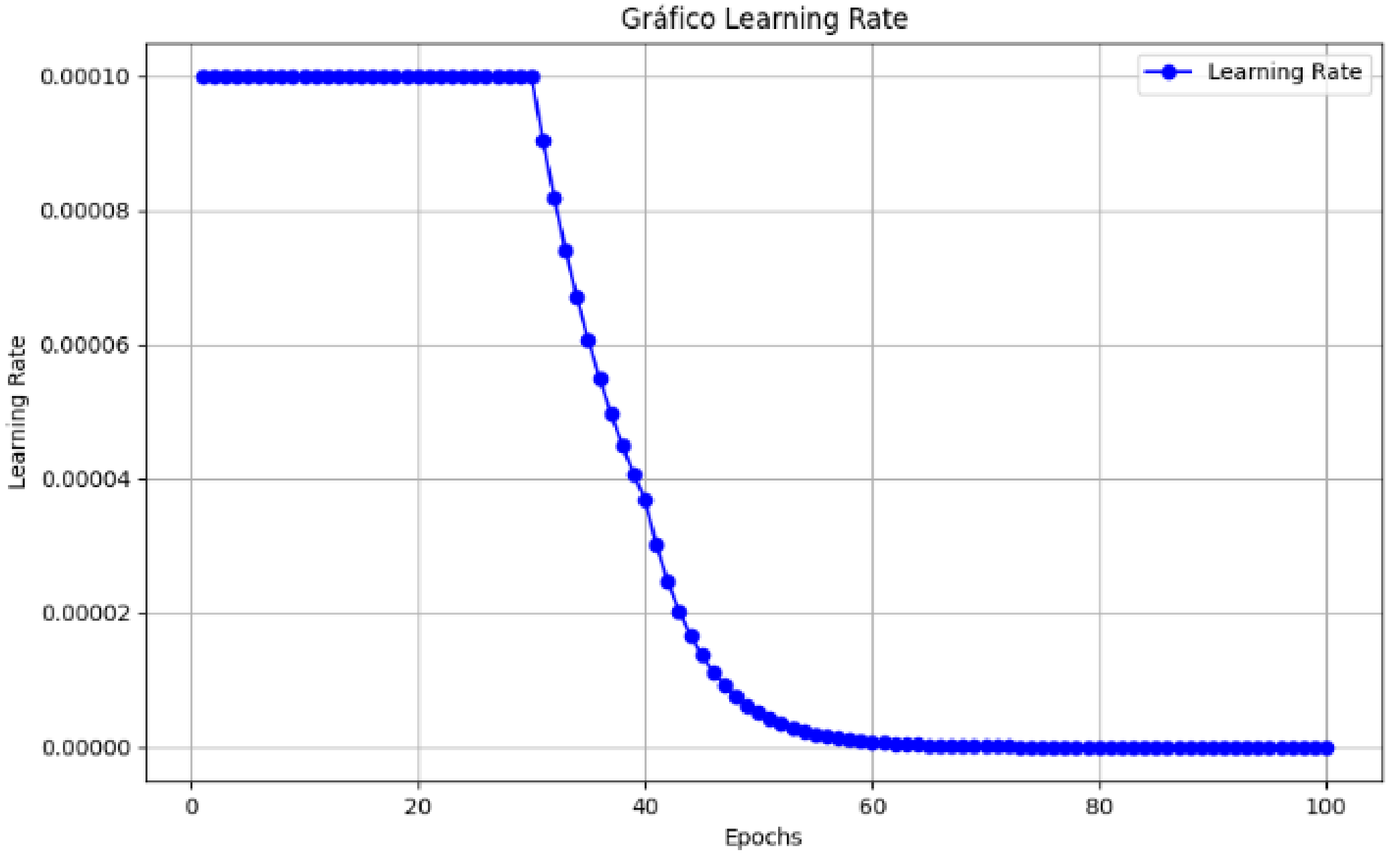
Pérdida durante el entrenamiento



Pérdida de validación durante el entrenamiento



Tasa de aprendizaje durante el entrenamiento



Texto real

```
print('~'*100, 'REAL TEXT')
[tf.strings.reduce_join([num_to_char(word) for word in sentence]) for sentence in [sample[1]]]
```

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ REAL TEXT
[<tf.Tensor: shape=(), dtype=string, numpy=b'place red at c six now'>]
```

Predicción

```
print('~'*100, 'PREDICTIONS')
[tf.strings.reduce_join([num_to_char(word) for word in sentence]) for sentence in
```

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ PREDICTIONS
[<tf.Tensor: shape=(), dtype=string, numpy=b'place red at c six now'>]
```

Texto real

```
print('~'*100, 'REAL TEXT')
[tf.strings.reduce_join([num_to_char(word) for word in sentence]) for sentence in [sample[1]]]
```

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ REAL TEXT
[<tf.Tensor: shape=(), dtype=string, numpy=b'set blue in a two please'>]
```

Predicción

```
print('~'*100, 'PREDICTIONS')
[tf.strings.reduce_join([num_to_char(word) for word in sentence]) for sentence in
```

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ PREDICTIONS
[<tf.Tensor: shape=(), dtype=string, numpy=b'set blue in a two please'>]
```

Texto real

```
print('~'*100, 'REAL TEXT')
[tf.strings.reduce_join([num_to_char(word) for word in sentence]) for sentence in
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ REAL TEXT
[<tf.Tensor: shape=(), dtype=string, numpy=b'lay blue in d four please'>]
```

Predicción

```
print('~'*100, 'PREDICTIONS')
[tf.strings.reduce_join([num_to_char(word) for word in sentence]) for sentence in
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ PREDICTIONS
[<tf.Tensor: shape=(), dtype=string, numpy=b'lay blue in d four please'>]
```

Texto real

```
print('~'*100, 'REAL TEXT')
[tf.strings.reduce_join([num_to_char(word) for word in sentence]) for sentence in
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ REAL TEXT
[<tf.Tensor: shape=(), dtype=string, numpy=b'bin red at s nine again'>]
```

Predicción

```
print('~'*100, 'PREDICTIONS')
[tf.strings.reduce_join([num_to_char(word) for word in sentence]) for sentence in
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ PREDICTIONS
[<tf.Tensor: shape=(), dtype=string, numpy=b'bin red at s nine again'>]
```

Análisis

¿POR QUÉ FUNCIONA LA SOLUCIÓN ?

1. **Capacidad espacio temporal:** Convoluciones 3D para extraer tanto características espaciales como temporales en el habla visual.

2. **Modelado secuencial con LSTM Bidireccionales:** Aprendizaje de contextos en ambas direcciones temporales.

3. **Robustez frente a la variabilidad:** Función de pérdida CTC para robustez en ausencia de segmentación perfecta.

4. **Reducción de sobreajuste:** Dropout post-LSTM para mitigar sobreajuste.

5. **Eficiencia en el aprendizaje:** Optimizador Adam con tasa de aprendizaje estable para convergencia eficiente.

Contexto

Este enfoque visual ofrece **nuevas posibilidades** al convertir el habla en texto, especialmente beneficioso para la accesibilidad de personas con **discapacidad auditiva**, prometiendo **autonomía y participación en contenido multimedia.**

Alcance

Entrenamiento intensivo en datos seleccionados busca mejorar el reconocimiento y transcripción del modelo, con visión a **convertirse en líder en transcripción asistida por visión** para desafíos más amplios en el futuro.

Análisis

¿POR QUÉ SE UTILIZÓ LA RED IMPLEMENTADA PARA EL PROBLEMA Y NO OTRA?

Adecuación a la tarea

Proyecto diseñado específicamente para el reconocimiento del habla visual.

Estado del arte

Basado en el modelo LipNet, que representaba el estado del arte en reconocimiento de habla visual.

Flexibilidad y generalización

Arquitectura flexible y capaz de generalizar a nuevos datos.

Disponibilidad de datos

Diseñado para trabajar con datos de video, cada vez más disponibles y ricos en información espacial y temporal.

Balance entre rendimiento y complejidad

Análisis

COSAS QUE SE PUEDEN HACER MEJOR CON MÁS TIEMPO Y RECURSOS:

Optimización del Conjunto de Datos

Ampliación del Conjunto de Datos y Escalabilidad

Optimización de Recursos Computacionales

Desarrollo de Interfaz y Adaptabilidad

SUGERENCIAS PARA FUTURAS ITERACIONES:

El modelo funciona según lo esperado, pero se podría aplicar early stopping para ahorrar tiempo de entrenamiento

Amplia documentación y modelos preentrenados disponibles en comunidades relacionadas con la lectura de labios.

Early stopping basado en métricas como costo, pérdida de validación y tasa de aprendizaje evitaría entrenamiento innecesario.

Sugerencia de utilizar fine tuning: congelar capas necesarias y entrenar con datos propios para adaptar el modelo a nuevas tareas o dominios específicos

CONCLUSIONES

1. El avance tecnológico, aplicado de manera adecuada, puede generar herramientas que mejoren la vida de personas con capacidades diferentes.

2. El modelo cumplió con las expectativas al "leer" los labios en videos, predecir mensajes y generar texto preciso.

3. Métricas como pérdida, pérdida de evaluación y tasa de aprendizaje indican un desempeño excelente del modelo, validando su correcta construcción.

4. Con más recursos y un conjunto de datos extenso, el modelo podría mejorar su rendimiento para aplicaciones cotidianas.

5. Decisiones correctas en la construcción del modelo, evidenciadas por resultados esperados y un cumplimiento efectivo de la tarea de lectura labial.

BIBLIOGRAFÍA

Chung, J. S. , et al. "Lip Reading Sentences in the Wild." 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) IEEE, 2017.