



Tecnológico
de Monterrey

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS
SUPERIORES DE MONTERREY

INTELIGENCIA ARTIFICIAL AVANZADA PARA LA
CIENCIA DE DATOS II

Momento de Retroalimentación

Módulo 2: Implementación de un modelo de Deep Learning

Autor:

Carlos de Jesús Ávila González A01750220

Repositorio:

https://github.com/Carlos1300/Retro_B2_M2

Profesor:

Julio Guillermo Arriaga Blumenkron

Fecha:

03/11/2022

1. Redes Neuronales Convolucionales

Las redes neuronales convolucionales (CNN) son un tipo de red neuronal artificial que se inspira de las redes neuronales de la corteza visual primaria de un cerebro biológico. Este tipo de red se puede considerar como una variación del perceptrón multicapa con la diferencia de que esta arquitectura hace uso de operaciones con matrices bidimensionales, lo que le brinda la capacidad de realizar operaciones con imágenes, la más popular de estas siendo la clasificación.

2. Problemática

Muchas veces las emociones de las personas se pueden llegar a confundir gracias a la capacidad que tienen algunos seres humanos para convencer a la gente y transmitirles una sensación. Sin embargo, la única manera de poder entender lo que una persona siente en un momento preciso es observar las pequeñas acciones que hacen ya sea con las manos, los pies o su propio rostro.

Con el objetivo de dar un paso más a la comprensión emocional de las personas y saber realmente qué es lo que están sintiendo con solo mirarlas fijamente y poner atención a sus expresiones faciales se elaborará una red neuronal convolucional que tenga la capacidad de analizar estas expresiones y decirnos, con cierta precisión, lo que está sintiendo la persona.

3. Recopilación de los Datos

La base de datos a utilizar fue extraída de la plataforma *Kaggle*, lleva como nombre **Emotion Detection** y consta de 35,685 imágenes en escala de grises que pertenecen a 7 clases (felicidad, neutral, tristeza, enojo, sorpresa, desagrado y miedo). La base de datos está dividida en dos carpetas, train con 28,709 imágenes y test con 7,178 imágenes.

4. Características principales de los modelos

Las características base que se aplicarán para todos los modelos se describen en la siguiente tabla:

Característica	Descripción
Tamaño de Input	(48, 48, 1)
Tamaño de Batch	64
Set de entrenamiento	20,099 imágenes
Set de validación	2,151 imágenes
Set de pruebas	7,178 imágenes
Clases	7 clases

Tabla 1: Características generales de los modelos

Para cada uno de los modelos se utilizará la siguiente configuración:

Configuración	Descripción
Función de pérdida	Categorical Crossentropy
Optimizador	Adam
Métrica	Accuracy
Número de épocas	10

Tabla 2: Configuraciones generales de los modelos.

5. Modelo base-line

El primer modelo generado cuenta con la siguiente arquitectura:

Tipo de Capa	Descripción
Input	(48, 48, 1)
Convolutional2D	32 filtros, tamaño de kernel de 3 y activación relu
MaxPool2D	tamaño de filtro de 2
Convolutional2D	64 filtros, tamaño de kernel de 3 y activación relu
MaxPool2D	tamaño de filtro de 2
Convolutional2D	128 filtros, tamaño de kernel de 3 y activación relu
MaxPool2D	tamaño de filtro de 2
Convolutional2D	256 filtros, tamaño de kernel de 3 y activación relu
MaxPool2D	tamaño de filtro de 2
Flatten	-
Dense	7 neuronas y activación softmax

Tabla 3: Arquitectura del Modelo Base-line.

5.1. Desempeño

Después de entrenar el modelo se obtuvo una precisión en el set de entrenamiento de un **67.44 %**, sin embargo en el set de validación obtuvimos un **56.02 %** lo que claramente nos indica que nuestro modelo se está sobre entrenando a los datos de entrenamiento. A continuación se muestran dos gráficas, a la derecha podemos observar la precisión en el set de entrenamiento y el de

validación a lo largo de las épocas; mientras que en la gráfica de la izquierda podemos ver el error en el set de entrenamiento y el de validación.

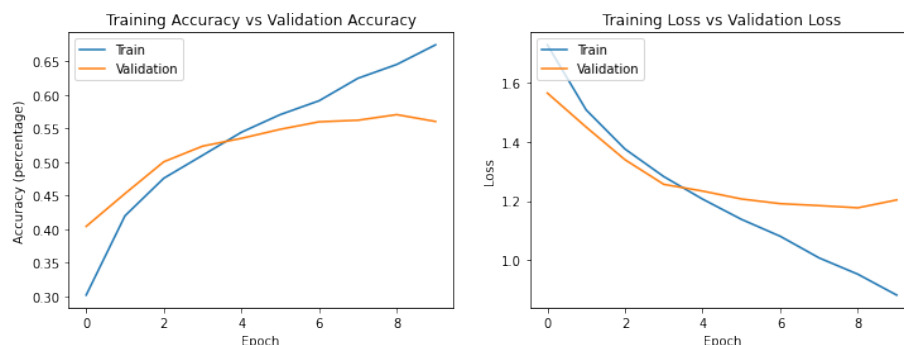


Figura 1: Desempeño del Modelo Base-line.

Al momento de evaluar el modelo con un set de pruebas este obtuvo un **55.0%** de precisión. Este porcentaje se intentará incrementar en las siguientes secciones utilizando diversas técnicas.

6. Modelo con Capas Densas

Este modelo aplica dos capas densas después de nuestra capa de flatten con el objetivo de mejorar el desempeño pasado. A continuación, se muestra una tabla donde se explica su arquitectura.

Tipo de Capa		Descripción
Input		(48, 48, 1)
Convolutional2D	32 filtros, tamaño de kernel de 3 y activación relu	
MaxPool2D	tamaño de filtro de 2	
Convolutional2D	64 filtros, tamaño de kernel de 3 y activación relu	
MaxPool2D	tamaño de filtro de 2	
Convolutional2D	128 filtros, tamaño de kernel de 3 y activación relu	
MaxPool2D	tamaño de filtro de 2	
Convolutional2D	256 filtros, tamaño de kernel de 3 y activación relu	
MaxPool2D	tamaño de filtro de 2	
Flatten		-
Dense	64 neuronas y activación relu	
Dense	32 neuronas y activación relu	
Dense	7 neuronas y activación softmax	

Tabla 4: Arquitectura del Modelo con Capas Densas

6.1. Desempeño

Posterior a la etapa de entrenamiento, esta arquitectura logró una precisión de **55.24 %** en el set de entrenamiento; mientras que en el de validación obtuvo un **53.23 %**. Esto nos indica que el modelo no está sobre entrenado, por lo que debería tener un buen desempeño en el set de pruebas.

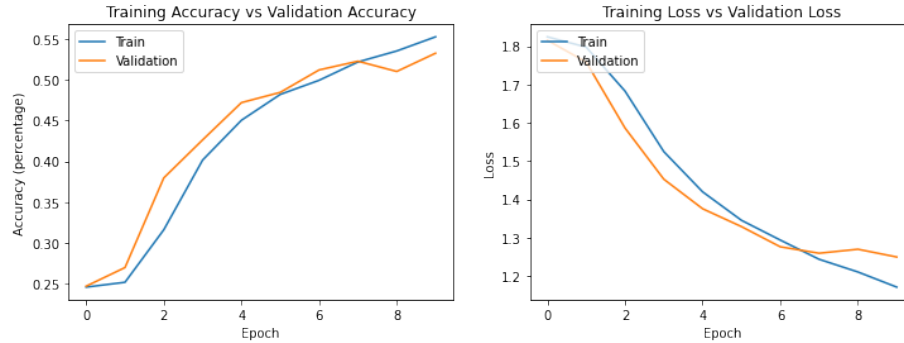


Figura 2: Desempeño del Modelo con Capas Densas.

Observando la gráfica nos podemos dar cuenta de que el modelo no se encuentra sobre entrenado. Sin embargo, su desempeño en el set de pruebas fue de un **51.4 %** de precisión. Comparado con el resultado anterior podemos decir que si bien se desempeña un poco peor, este es capaz de generalizar de mejor manera los datos.

7. Modelo con Batch Normalization

Con el objetivo de mejorar el desempeño del modelo base-line se elaborará otro modelo que tenga en su arquitectura diversas capas de batch normalization. La arquitectura es la siguiente:

Tipo de Capa	Descripción
Input	(48, 48, 1)
Convolutional2D	256 filtros, tamaño de kernel de 3 y activación relu
BatchNormalization	-
MaxPool2D	tamaño de filtro de 2
Convolutional2D	128 filtros, tamaño de kernel de 3 y activación relu
BatchNormalization	-
MaxPool2D	tamaño de filtro de 2
Convolutional2D	64 filtros, tamaño de kernel de 3 y activación relu
BatchNormalization	-
MaxPool2D	tamaño de filtro de 2
Convolutional2D	32 filtros, tamaño de kernel de 3 y activación relu
BatchNormalization	-
MaxPool2D	tamaño de filtro de 2
Flatten	-
Dense	7 neuronas y activación softmax

Tabla 5: Arquitectura del Modelo con Batch Normalization

7.1. Desempeño

En la etapa de entrenamiento el modelo obtuvo un **76.14 %** de precisión en el set de entrenamiento y en el de validación un **48.16 %** lo que nos indica que el modelo está sumamente sobre entrenado. Las gráficas resultantes se muestran a continuación.

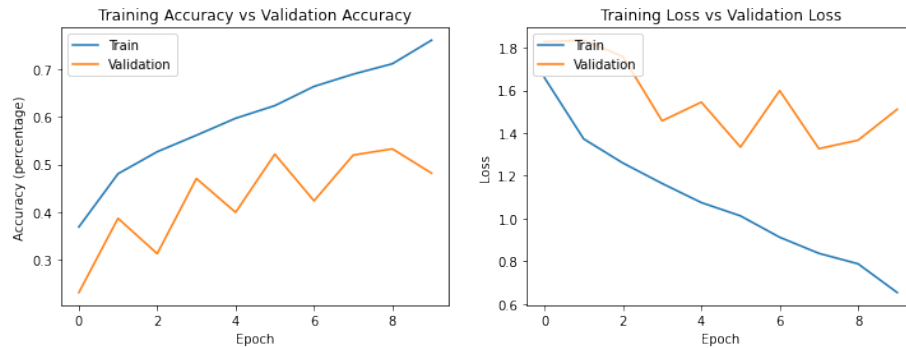


Figura 3: Desempeño del modelo con Batch Normalization.

En la etapa de evaluación, el modelo obtuvo un **47.4 %** de precisión, Lo que está bastante alejado de los dos resultados que hemos obtenido hasta el momento.

8. Modelo con Batch Normalization y Dropout

La última arquitectura que se probará para intentar mejorar el desempeño del modelo es la combinación de Batch Normalization con diversas capas de Dropout. Esta arquitectura es definida en la siguiente tabla.

Tipo de Capa		Descripción
Input		(48, 48, 1)
Convolutional2D	256 filtros, tamaño de kernel de 3 y activación relu	
BatchNormalization		-
MaxPool2D		tamaño de filtro de 2
Dropout		25 % de apagar la neurona
Convolutional2D	128 filtros, tamaño de kernel de 3 y activación relu	
BatchNormalization		-
MaxPool2D		tamaño de filtro de 2
Convolutional2D	64 filtros, tamaño de kernel de 3 y activación relu	
BatchNormalization		-
MaxPool2D		tamaño de filtro de 2
Dropout		25 % de apagar la neurona
Convolutional2D	32 filtros, tamaño de kernel de 3 y activación relu	
BatchNormalization		-
MaxPool2D		tamaño de filtro de 2
Flatten		-
Dense	7 neuronas y activación softmax	

Tabla 6: Arquitectura del Modelo con Batch Normalization

8.1. Desempeño

Esta arquitectura tuvo un **61.37 %** de precisión en el set de entrenamiento y en el de validación un **42.31 %** de precisión. De nuevo podemos observar que el modelo está sobre entrenado. La gráfica donde se muestra su desempeño es la siguiente:

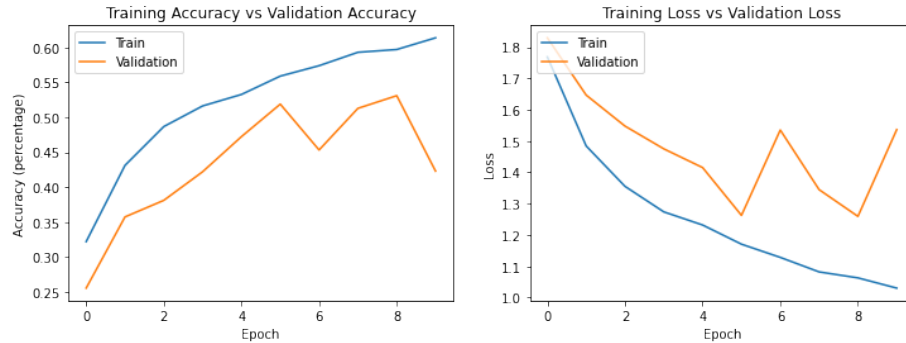


Figura 4: Desempeño del modelo con Batch Normalization y Dropout.

Al probar esta arquitectura en el set de pruebas se obtuvo un **43.0 %** de precisión, siendo esta la que menos porcentaje de precisión obtuvo en las pruebas.

9. Análisis de Resultados

A continuación se muestra un ranking de las arquitecturas y sus resultados en el set de pruebas:

Arquitectura	Precisión
Modelo Base-line	55.0 %
Modelo con Capas Densas	51.4 %
Modelo con Batch Normalization	47.4 %
Modelo con Batch Normalization y Dropout	43.0 %

Tabla 7: Ranking de las arquitecturas.

El Modelo Base-line fue el que mejor precisión obtuvo en esta ocasión, sin embargo, pudimos observar que este tenía tendencia a caer en un sobre entrenamiento. Este problema de sobre entrenamiento fue solucionado por el Modelo con Capas Densas, lo que nos deja como conclusión que si quisiéramos mejorar aún más estas arquitecturas una buena forma de comenzar sería tomar los primeros dos modelos y aplicarles distintas técnicas de regularización.

10. Trabajo a Futuro

- Integración de nuevas técnicas tomando como base el Modelo con Capas Densas resultante de nuestras pruebas realizadas.
- Introducción de un callback para poder detener el entrenamiento cuando el modelo empiece a tener cierta tendencia a sobre entrenarse.

- Integración de Transfer Learning con diversas modificaciones realizadas por nosotros para solucionar nuestra problemática principal.