Fundamentos de deep learning - Sign Language Recognition MLP vs CNN

Departamento de Ingeniería de Sistemas,
Universidad de Antioquia, Colombia

Abstract— Las actividades propuestas para la entrega del proyecto del curso, buscan que cada uno de los grupos de estudiantes presenten todo el diseño, análisis y simulación de un sistema de predicción basado en técnicas de aprendizaje profundo; describiendo el problema y su contexto en términos del estado del arte, especificando cada una de las etapas del desarrollo del trabajo, los modelos con sus respectivas restricciones, la metodología de validación, los resultados y conclusiones obtenidas.

Index Terms— Clasificación multiclase, aprendizaje profundo, procesamiento de imágenes.

I. INTRODUCCIÓN

El lenguaje de señas es una forma de comunicación que utilizan las personas sordas o con problemas auditivos para interactuar con su entorno social. Se basa en la producción y percepción de gestos que tienen un significado lingüístico y que siguen una estructura gramatical definida. El lenguaje de señas no es universal, sino que existen diferentes lenguas de señas en el mundo, cada una con su propio vocabulario y reglas.

Un problema de clasificación multiclase consiste en asignar una etiqueta a un objeto o situación a partir de un conjunto de posibles opciones. Por ejemplo, clasificar una imagen de un animal en una de las categorías: perro, gato, pájaro, etc. En este caso, el objetivo es clasificar imágenes de señas en alguna de las letras del alfabeto del lenguaje de señas.

Este proyecto tiene una gran utilidad social, ya que podría facilitar la comunicación entre las personas sordas y las oyentes, así como el aprendizaje y la enseñanza del lenguaje de señas. Además, podría contribuir al reconocimiento y la difusión de las lenguas de señas como lenguas naturales y culturales de las comunidades sordas.

Para abordar este problema se cuenta con un dataset que contiene 17.111 imágenes de gestos de las letras del alfabeto en el lenguaje de señas inglés. El objetivo es reconocer los gestos y clasificarlos en una de las 27 clases posibles. El formato del dataset es similar al del MNIST clásico, con una fila de cabecera que indica la etiqueta y los valores de los píxeles. Cada imagen tiene un tamaño de 28x28 píxeles y un valor de escala de grises entre 0 y 255. El dataset se basa en imágenes originales de diferentes usuarios realizando los gestos contra distintos fondos, que fueron procesadas y modificadas para aumentar la cantidad y la variedad de los datos.

Como se puede observar en la figura 1 (diagrama de barras), se está abordando un problema de clasificación balanceado, donde la cantidad de muestras por cada clase es aproximadamente la misma.

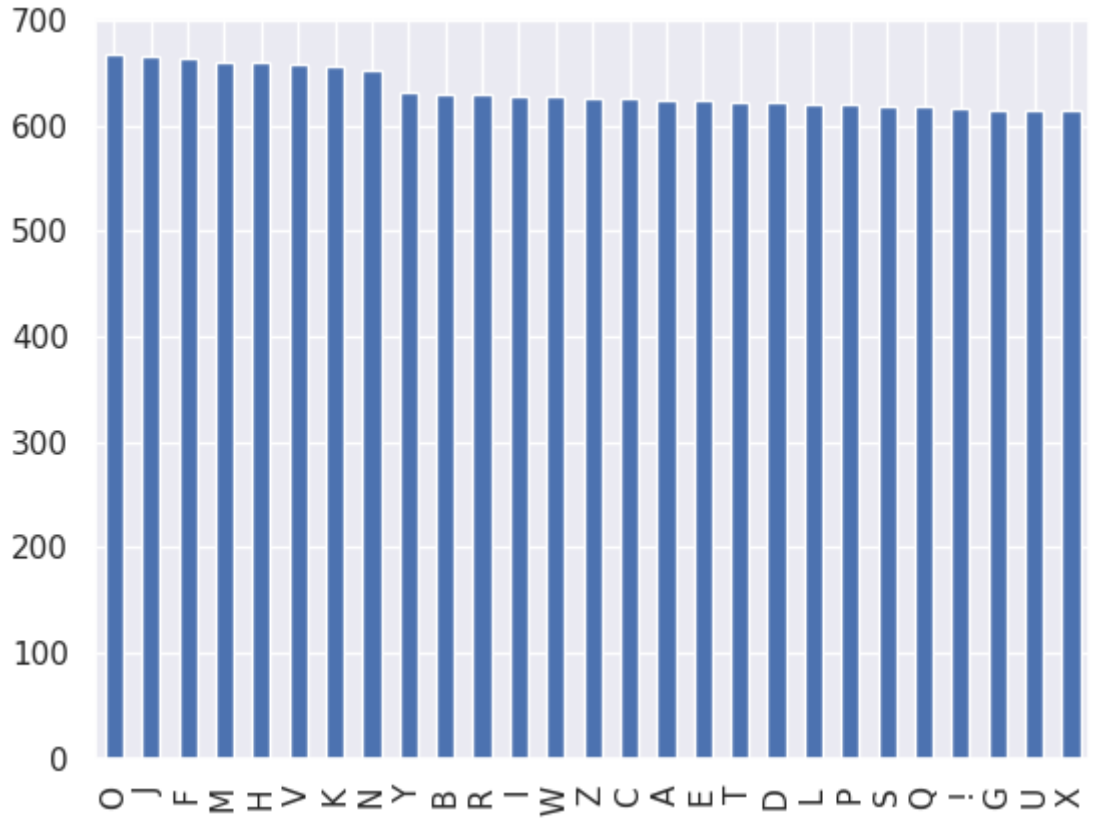

Figura 1. Diagrama de barras para ilustrar la cantidad de muestras por clase.

II. ESTADO DEL ARTE

Se han revisado cuatro artículos que abordan un problema similar al tratado en este informe:

- Reconocimiento de lenguaje de señas
- Traducción de lenguaje de señas a texto

American Sign Language Alphabet Recognition Using Microsoft Kinect [1]: Se describe una metodología para el reconocimiento del alfabeto de lenguaje de señas americano usando el Kinect de Microsoft (Para la generación del dataset). Con este enfoque, los autores lograron un accuracy promedio del 92%.

American Sign Language alphabet recognition using Convolutional Neural Networks with multiview augmentation and inference fusion [2]: Se aborda el problema de reconocimiento de las letras del alfabeto del lenguaje de señas americano haciendo uso de dos dataset públicos que se usan como benchmark, logrando un accuracy del 93% y 100% (Dependiendo de la metodología de validación usada).

Recognition of Arabic Sign Language Alphabet Using Polynomial Classifiers [3]: Se realiza la misma tarea pero tomando el lenguaje de señas árabe y haciendo uso de clasificadores polinomiales, además se comparan los errores de clasificación con modelos basados en modelos ANFIS obteniendo los siguientes resultados:

Error/modelo	ANFIS	Polinomiales	Reducción
Train data	41	26	36.6%
Test data	108	46	57.4%

Tabla 1. Resultados obtenidos en el paper Recognition of Arabic Sign Language Alphabet Using Polynomial Classifiers.

SIGN LANGUAGE ALPHABET RECOGNITION USING CONVOLUTION NEURAL NETWORK [4]: Se trabaja el problema de reconocimiento del alfabeto lenguajes de señas americano logrando un accuracy del 0.9963.

III. METODOLOGÍA

El dataset original viene particionado en dos conjuntos: Uno para entrenamiento y otro para validación, cada uno con 12844 y 4267 muestras respectivamente, el dataset de validación **NUNCA** se utiliza para el entrenamiento o selección de los hiper parámetros más adecuados para el modelo.

Cada feature del dataset corresponde a un píxel de la imagen, y dado que tenemos imágenes de 28x28 pixeles, tendremos un total de 784 píxeles.

Se estandarizaron los dataset con la estrategia Min-Max (Se ajustó el escalador con el dataset de entrenamiento, y con él se escaló también el dataset de validación).

Se usaron dos modelos:

- Feedforward neural network
- Convolutional neural network

Con variaciones en algunos de los hiper parámetros de cada uno de ellos.

Ahora bien, cabe notar que los conjuntos de entrenamiento y validación se redimensionaron para ser usadas en el modelo convolucional.

Dado el alto coste computacional que tiene el entrenamiento de las redes neuronales, del conjunto de entrenamiento se decidió extraer una porción para entrenamiento y otro para validación, esto podría entenderse como un k-fold de 1 solo fold.

IV. MODELACIÓN

Estos fueron los resultados obtenidos en cada uno de los modelos

Perceptrón multicapa			
Cantidad de capas ocultas	**Neuronas por capa**	**Accuracy training**	**Accuracy testing**
1	32	0.0399	0.0370
1	64	0.0399	0.0370
1	128	1.0000	0.9707
1	256	1.0000	0.9766
1	512	1.0000	0.9736
2	32	0.2449	0.2241
2	64	0.8308	0.7686
2	128	0.9998	0.9998
2	256	1.0000	0.9769
2	512	0.9866	0.9507
4	32	0.8331	0.7679
4	64	0.9947	0.9947
4	128	0.9608	0.9148

4	256	1.0000	0.9830
4	512	0.9943	0.9290
8	32	0.9147	0.8410
8	64	0.0399	0.0399
8	128	0.8015	0.7306
8	256	0.6390	0.5638
8	512	0.0399	0.0370
16	32	0.0399	0.0370
16	64	0.0399	0.0370
16	128	0.0399	0.0370
16	256	0.0399	0.0370
16	512	0.0399	0.0370

Tabla 2. Resultados obtenidos usando el modelo de perceptrón multicapa con diferentes hiper parámetros.

Redes neuronales convolucionales					
Con maxpooling	**Cantidad de capas convolucionales**	**Cantidad de filtros por capa**	**Tamaño del filtro por capa**	**Accuracy training**	**Accuracy testing**
True	3	32, 16, 8	7, 5, 3	0.9149	0.9672
True	4	64, 32, 16, 8	9, 7, 5, 3	0.0406	0.0370
True	5	128, 64, 32, 16, 8	11, 9, 7, 5, 3	0.0377	0.0354
False	3	32, 16, 8	7, 5, 3	0.9599	0.9686

Tabla 3. Resultados obtenidos usando el modelo de redes neuronales convolucionales con diferentes hiper parámetros.

Se logra evidenciar que en algunos casos el modelo se malogra con sobreajuste (Se distancia el accuracy de entrenamiento del accuracy de testing) como pasa en el perceptrón multicapa de 8 capas ocultas y 32 neuronas por capa (Ver tabla 2).

Tomaremos un modelo ganador de ambas tablas, y estudiaremos más a fondo qué tan bien se está realizando la clasificación analizando la matriz de confusión; en este caso, los modelos cuyos accuracies son altos y la resta de su accuracy de entrenamiento y de validación es baja (Que no presente overfitting):

- Perceptrón multicapa con 2 capas ocultas y 128 neuronas por capa oculta.
- Red neuronal convolucional con maxpooling, con 3 capas convolucionales, con 32, 16 y 8 filtros respectivamente y con tamaños de 7, 5 y 3

Para reducir el tiempo de entrenamiento (Costo computacional), se decidió reducir la cantidad de épocas (128 en los resultados anteriores) a 12.

A continuación, Figura 2, la matriz de confusión obtenida para el primer modelo:

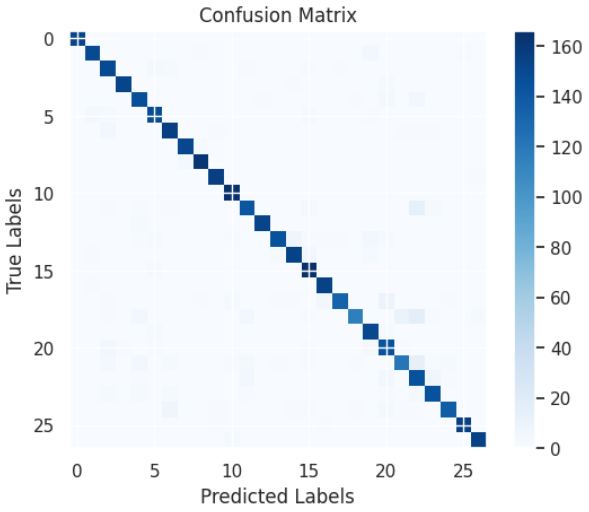

Figura 2. Matriz de confusión del perceptrón multicapa MLP(nlayers=2, lsize=128).

A continuación, Figura 3, la matriz de confusión obtenida para el segundo modelo:

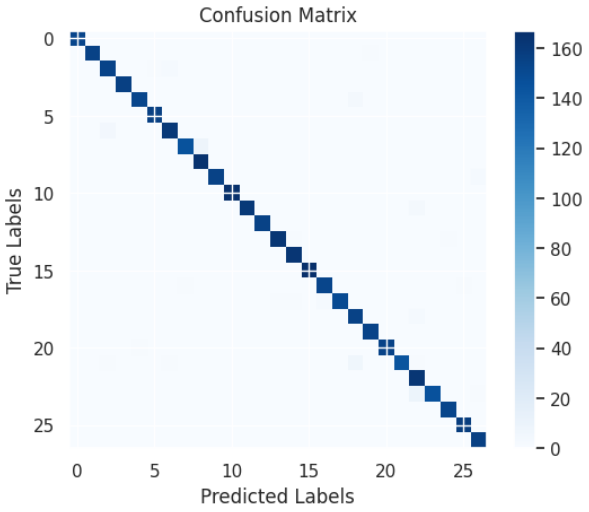

Figura 3. Matriz de confusión de la red neuronal convolucional.

Como era de esperar dados los resultados de las tablas 1 y 2, estos modelos tienen una buena capacidad de generalización, en el caso particular del MLP se presentan algunas impurezas por fuera de la diagonal principal (Equivocaciones) con respecto a la matriz del modelo convolucional.

VIII. PREDICCIONES

Además de las matrices de confusión, se calcula el accuracy obtenido en ambos casos para posteriormente hacer un contraste de resultados con los encontrados en el estado del arte:

	Accuracy test
Perceptrón multicapa	0.9379
Convolucional	0.9873

VIII. CONTRASTE DE RESULTADOS

Encontramos que nuestra solución es competitiva, ya que los papers reportan accuracies en el rango [92%,100%].

BIBLIOGRAFÍA

[1] Cao Dong, Leu, M. C., & Yin, Z. (2015). American Sign Language alphabet recognition using Microsoft Kinect. 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). doi:10.1109/cvprw.2015.7301347

[2] Tao, W., Leu, M. C., & Yin, Z. (2018). American Sign Language alphabet recognition using Convolutional Neural Networks with multiview augmentation and inference fusion. Engineering Applications of Artificial Intelligence, 76, 202–213. doi:10.1016/j.engappai.2018.09.006

[3] Assaleh, K., & Al-Rousan, M. (2005). Recognition of Arabic Sign Language Alphabet Using Polynomial Classifiers. EURASIP Journal on Advances in Signal Processing, 2005(13). doi:10.1155/asp.2005.2136

[4] Kumar, M., Gupta, P., Jha, R. K., Bhatia, A., Jha, K., & Shah, B. K. (2021). Sign Language Alphabet Recognition Using Convolution Neural Network. 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS). doi:10.1109/iciccs51141.2021.9432296