Tarea 3 – IIC3724

Benjamín Farías V.
Pontificia Universidad Católica de Chile

***Motivación*— La detección de enfermedades respiratorias mediante radiografías siempre ha sido un tema muy relevante en la medicina, y considerando el contexto de epidemia actual por el COVID19, se torna más importante que nunca. Un buen detector de dichas enfermedades en etapas tempranas puede resultar en miles de vidas salvadas.**

I. SOLUCIÓN PROPUESTA

La solución propuesta utiliza una **Red Neuronal Supervisada (Multi Layer Perceptron)** como modelo predictivo, la que cuenta con 2 capas ocultas de tamaño 50 y ocupa la función de activación Rectified Linear Unit (ReLU). Este tipo de clasificador funciona entrenando una red de capas de perceptrones conectados, donde cada uno recibe un input con pesos específicos y lo transforma en un output aplicando la función de activación. Para entrenar el modelo se pasan los datos por la red completa varias veces, ajustando los pesos de entrada a cada perceptrón de forma de minimizar el error con respecto al resultado esperado. Al final de este proceso la red habrá 'aprendido', y será capaz de realizar predicciones de alto rendimiento.

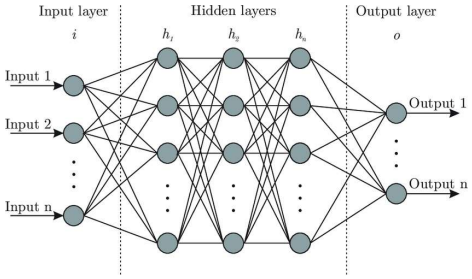

Fig. 1. Red Neuronal Supervisada (MLP)

Las características de interés extraídas de las imágenes en esta tarea fueron **LBP, Haralick, HoG y Gabor** (explicadas y justificadas en la tarea anterior). Tras la etapa de extracción de estas características, se les aplicó una función de limpieza (*clean*) y normalización, con el objetivo de eliminar redundancia y estandarizar los datos. A continuación, se utilizaron diversas estrategias ocupando combinaciones entre *SFS* (criterio de *Fisher*) y *Mutual Information* como selectores de características, y *PCA* como transformador de características, para finalmente entregarle los datos al clasificador (todo esto se detalla en la sección Experimentos).

II. EXPERIMENTOS REALIZADOS

Previo a probar las 5 estrategias, se obtuvieron los siguientes resultados mediante experimentación con *SFS* y varios clasificadores (*k-NN, LDA, DMin, SVM, RForest, AdaBoost, NN*):

- *SFS* funciona **muy bien** entregando 24 características
- Los **mejores 3 clasificadores** para este problema son *SVM* (lleva los datos a un espacio en el que son separables mediante planos, gracias a la aplicación de una función de *kernel*), *Neural Network* (explicada en la sección anterior) *y Random Forest* (entrena un conjunto de árboles de decisión escogiendo las características de separación de forma semi-aleatoria, para finalmente obtener la mayoría de los votos).
- La ***SVM* óptima** tiene su parámetro C igual a 1 y utiliza el kernel ***RBF***. La ***NN* óptima** se describe en la sección anterior. El ***Random Forest* óptimo** contiene 1000 estimadores y utiliza el criterio de **entropía.**

Sabido esto, se probaron las **siguientes 5 estrategias**:

- ***MI + SFS:*** Se aplica *Mutual Information* a las características (se encarga de cuantificar la información compartida entre los datos, para después eliminar las características más redundantes al estar en conjunto con las demás, de forma similar a la entropía). Tras esto se aplica *SFS* para obtener las 24 mejores características y se pasa el resultado por los 3 clasificadores mencionados más arriba **(3 estrategias en total, 1 por cada clasificador)**. Se obtuvo que el rendimiento empeora un poco en comparación a utilizar *SFS* por sí solo, además de que se descartó el *Random Forest* ya que tuvo un rendimiento un poco peor a los otros 2 clasificadores.
- ***SFS + PCA*:** Se aplica *SFS* para obtener las 50 mejores características, luego se transforman en 30 mediante *PCA* para a continuación volver a aplicar *SFS*, pero para 24 características. El resultado se pasa por los 2 clasificadores (*SVM y NN*) **(2 estrategias en total)**. Se obtuvo que el rendimiento también es peor, pero fue útil para encontrar que *NN* rinde más que *SVM* en este problema.

(Los detalles cuantitativos de estos experimentos se encuentran en el archivo adjunto *'Results.txt'*)

Con la información obtenida de los experimentos, se decidió aplicar la estrategia SFS_24 + PCA_20 + *NN*, con la intención de obtener las 24 características que se sabía **rendían muy bien**, y luego **mejorarlas al reducirlas a 20 componentes principales** (menor correlación). Esta combinación definitiva logró el 95.71% y 100% de **exactitud** en la clasificación de **patches y pacientes**, respectivamente **(preferir ver las imágenes adjuntas para apreciar mejor los números).**

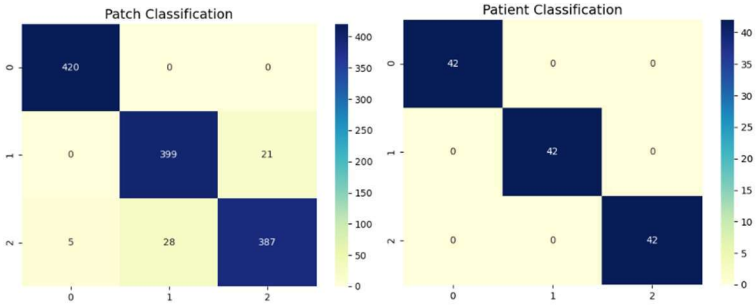

Fig. 2. Matrices de Confusión Finales

III. CONCLUSIONES

Podemos afirmar que, para este tipo de problemas complejos, las redes neuronales son una muy buena opción, ya que su estructura inspirada en el cerebro humano posee una gran capacidad de aprender de los datos, independiente de como estos distribuyan. Para problemas más simples convendría utilizar otros clasificadores menos complejos, tales como k-NN, ya que evitaría una implementación más compleja y costosa de lo necesario.