

Tarea 8
IELE-4017 Análisis Inteligente de Señales y Sistemas
Profesor: Luis Felipe Giraldo Trujillo
2021-I

En esta tarea no pueden utilizar funciones predefinidas para evaluar la función logística, realizar la regresión logística, hacer automáticamente particiones de los conjuntos de entrenamiento y validación, realizar la clasificación, calcular errores de clasificación, calcular matrices de confusión. Alguna falta de este tipo en cualquiera de los enunciados de esta tarea implicará que la tarea completa tenga una calificación de 0.0.

1. (40 puntos) **Clasificación de expresiones faciales utilizando regresión logística.** La falta de la solución en alguno de los enunciados de esta tarea implicará una nota de 0.0 en este ejercicio.

La carpeta `faces` contiene 974 imágenes de personas posando expresiones faciales y los respectivos landmarks (n landmarks por cada expresión facial). Recuerde que cada landmark es una coordenada en dos dimensiones. La penúltima letra del nombre de cada archivo indica qué tipo de expresión facial la persona está posando. La notación es la siguiente: `n` para neutral, `h` para feliz, `s` para triste, `a` para furioso, `d` para disgusto, y `f` para miedo. Sólo vamos a utilizar los landmarks en este ejercicio. Las imágenes son para referenciar los landmarks. El objetivo en este ejercicio es entrenar un clasificador que aprenda a clasificar una expresión facial de felicidad del resto de las expresiones faciales de otras emociones. **Vamos a asumir que la expresión de la emoción felicidad es clase 1 y las expresiones faciales del resto de las emociones es clase 0.**

- Del conjunto total de imágenes, saque sólo las imágenes con sus landmarks de la emoción felicidad. Este sería el conjunto de la clase 1, y sea N_h el tamaño de este conjunto. Ahora, del resto de imágenes de todas las otras emociones, saque un subconjunto de tamaño N_h de tal manera que el número de muestras de cada emoción esté lo más balanceado posible. Es decir, la cantidad de imágenes de expresiones neutrales, tristes, furiosas, disgustadas, y miedosas cada una debe ser lo más aproximado a $N_h/5$. Este conjunto sería el conjunto de la clase 0.
- Divida el conjunto de la clase 1 y la clase 0 en dos particiones: conjunto de entrenamiento con 80 % de los datos y conjunto de validación con 20 %. Trate de que las clases estén balanceadas en ambas particiones.
- Con sólo los datos de entrenamiento de ambas clases**, calcule la media de Procrustes de los landmarks. Es una media de Procrustes para todos los datos de entrenamiento, y no una media para cada clase. Utilizando esta media de Procrustes, alinee los landmarks en las particiones de entrenamiento y validación. Este paso sería el de preprocesamiento (es decir, no hay necesidad de estandarizar los datos).
- Con los landmarks alineados, cree un vector de características de tamaño 134 por cada imagen de entrenamiento y validación. La primera característica corresponde a la componente horizontal del landmark 1, y la segunda característica corresponde a la componente vertical del landmark 1 de la expresión facial. Repita este procedimiento para el resto de los landmarks.
- Identifique los índices de las características (en total 134) que están asociadas a: ojo izquierdo, ojo derecho, ceja izquierda, ceja derecha, boca, nariz, y contorno de la cara.
- Clasificación a través de regresión logística:** Implemente una rutina que, con los datos de entrenamiento, aprenda un clasificador lineal que discrimine las expresiones de felicidad del resto de las emociones (clase 1 y clase 0). **No pueden utilizar funciones preestablecidas para aprender un modelo de regresión logística, pero sí pueden utilizar un toolbox de optimización para resolver la minimización de la entropía cruzada. Usted, a mano, tiene que definir la función objetivo a minimizar.** Sea $w_h = [w_0, w_{h1}, w_{h2}, \dots, w_{h134}]^T$ el vector que define el clasificador lineal estimado. Recuerde que cada

elemento de w_h está asociado a una característica (excepto w_0 , que es el bias), indicando la relevancia de la característica en la clasificación.

- g) Con la rutina del enunciado f), calcule el error de clasificación sobre el conjunto de validación para al menos 10 valores de $\lambda > 0$. Grafique una curva con los valores de error versus λ . Analice los resultados obtenidos, e identifique el valor de λ que produce menor error de clasificación en el conjunto de validación.
- h) Con el valor de λ seleccionado en el enunciado g), calcule la matriz de confusión en los datos de validación. Comente los resultados obtenidos.
- i) Utilizando una sola gráfica de barras, grafique cada pareja $(j, |w_{hj}|)$ (es decir, el eje horizontal de la gráfica es el índice de la característica y el eje vertical es la magnitud del elemento del vector), $j = 1, \dots, 134$. Identifique en la gráfica las componentes de la cara a las que pertenece cada característica. Comente los resultados obtenidos e interprete el clasificador encontrado por el proceso de aprendizaje automático.

2. (60 puntos) **Mini red convolucional para clasificación de dígitos.** Los archivos `images2.mat` e `images9.mat` contienen cada uno 100 imágenes 14×14 reales de dígitos escritos del número 2 y del número 9, respectivamente. Estos hacen parte de la famosa base de datos llamada MNIST. El objetivo en este ejercicio es encontrar una red convolucional pequeña que sea capaz de clasificar una imagen como dígito 2 o dígito 9. En este ejercicio no vamos a realizar particiones de los datos para facilitar el análisis. Todo el conjunto de entrenamiento es el mismo de validación. Asumamos que el dígito 2 es de clase 1 y que el dígito es de clase 0.

- a) Construya una rutina que reciba una imagen de 14×14 denotada por I , un vector de tamaño 9 denotado por h , y un vector con 197 posiciones ($14 \times 14 + 1$) denotado por w , y realice el siguiente procedimiento:
 - Tome la imagen I , y realice un filtrado utilizando un filtro lineal de 3×3 cuyos coeficientes son los valores de h . Para realizar la convolución con el filtro puede utilizar la función `imfilter` de Matlab o su equivalente en Python). La imagen filtrada se denota por I_m .
 - Después tome la imagen I_m , la vectorice resultando en un vector de tamaño 196, y la pase por una red neuronal con una sola neurona y función de activación dada por una función logística. Los pesos de la red están dados por w . Es decir, si v es la imagen I_m vectorizada, la salida de la red es

$$\hat{y} = \frac{1}{1 + \exp^{w_0 + \sum_{j=1}^{196} v_j w_j}}. \quad (1)$$

Note que \hat{y} estaría entre 0 y 1.

Este modelo lo llamaremos una mini red convolucional, y la operación la denotaremos como $\hat{y} = g(I; w, h)$ (es decir, una función que recibe I como entrada, tiene parámetros w y h , y devuelve \hat{y}).

- b) Implemente una rutina que encuentre los pesos w y coeficientes h tal que la salida de la red \hat{y} tome un valor de 1 para una imagen de el dígito 2 y tome un valor de 0 para una imagen de el dígito 9. Utilice el criterio de error cuadrático. En particular, resuelva el problema de minimizar la siguiente función de costo:

$$E(w, h) = \left[\sum_{i=1}^{200} (y_i - g(I_i; w, h))^2 \right] + \lambda \| [w, h]^T \|_2^2,$$

donde $y_i = 1$ si la imagen I_i es del dígito 2 y $y_i = 0$ si la imagen I_i es del dígito 9. El parámetro $\lambda > 0$ define el grado de regularización. Se recomienda utilizar de condición inicial para h en la optimización los coeficientes de un filtro identidad, es decir, un filtro cuyos coeficientes son 0 excepto el coeficiente de la mitad, que toma un valor de 1.

- c) Con la rutina del enunciado b), calcule el error de clasificación sobre el conjunto de entrenamiento para al menos 5 valores de $\lambda > 0$. Grafique una curva con los valores de error versus λ . Analice los resultados obtenidos, e identifique el valor de λ que produce menor error de clasificación en el conjunto de entrenamiento.
- d) Analice los resultados obtenidos. En particular:
- Indique cuántas veces se clasificó mal/bien para cada clase. Muestre algunas imágenes que se clasifican bien y otras que se clasifican mal en cada clase.
 - Grafique el filtro resultante, es decir, el filtro de 3×3 que se construye con h . Píntelo utilizando la función *imagesc* o *bar3* de Matlab o sus equivalentes en Python. Muestre algunas imágenes que se clasifican bien de ambos dígitos (2 y 9) después de pasar por este filtro. Basado en estas imágenes, y en los coeficientes del filtro, determine qué tipo de operación hace este filtro.
 - Utilizando la función *imagesc* o *bar3* de Matlab o su equivalente en Python, grafique **el valor absoluto** de los pesos en w en forma de imagen de 14×14 consistente con la imágenes de los dígitos. Es decir, el peso en w que multiplica la posición (2, 3) de la imagen del dígito debería pintarse en la posición (2, 3). Con esta gráfica del valor absoluto de las componentes de w , determine la región o regiones de las imágenes filtradas sobre las cuales la red neuronal en la ecuación (1) se enfoca para hacer la clasificación.