Departamento de Ingeniería Matemática
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
UNIVERSIDAD DE CHILE

Curso: Aprendizaje de Máquinas, MA5204
Profesor: Felipe Tobar
Profesores Auxiliares: V. Faraggi, F. Fêtis, B. Moreno, F. Vásquez, A. Wortsman
Fecha de publicación: 12 de mayo

Tarea Práctica #2

Dedicación recomendada: 10 horas

Fecha de entrega: 2 de junio

Instrucciones: La tarea es grupal, en grupos de 2 o 3 integrantes, ni más ni menos. En caso de ser un grupo de 2 personas, deben realizar P1, en el caso de 3 personas, deben realizar P1 y P2. El formato de entrega es un reporte en `.pdf` de máximo 2 planas sin anexos, con doble columna, abstract, título e integrantes. Si el grupo es de 3 personas, son 3 planas máximo. Si hace su reporte en LaTeX, utilice el template de la conferencia ICML disponible en este enlace, o bien uno de formato similar. También debe entregar los códigos utilizados en formato `.html`, los cuales puede obtener descargando directamente desde *Jupyter* su notebook en ese formato.

P1. Handwritten Digits Classification

El objetivo de este problema es extender el algoritmo del perceptrón para clasificación binaria, a uno de K-clases utilizando el enfoque *One versus All*. Para ello, consideraremos el dataset *sklearn handwritten digits* cuya información se puede encontrar aquí *load digits*.

1. Cargue el dataset utilizando las siguientes lineas de código, se incluye además una visualización de los datos.

```
from sklearn.datasets import load_digits
import matplotlib.pyplot as plt
X, y = load_digits(return_X_y=True)
plt.imshow(X[0].reshape(8,8), cmap=plt.cm.gray_r, interpolation='nearest')
```

   Realice una exploración del dataset, cree una función que permita visualizar un dato y la clase correspondiente, muestre ejemplos de distintas clases.

2. Cree una clase[1] Perceptron() que implemente el algoritmo del Perceptrón y que discrimine un dígito de todo el resto. Su clase debe recibir al menos los siguientes parámetros:

   - **dig :** El dígito que se ha de clasificar. (int)
   - **lr:** El learning rate utilizado para el descenso de gradiente estocástico. (float)
   - **max_iter:** La cantidad máxima de iteraciones del algoritmo. (int)
   - **tol:** Criterio de detención del algoritmo. (float)

   Recuerde inicializar los pesos aleatoriamente, se solicita además que su clase tenga al menos los siguientes métodos:

[1] Le puede ser útil visitar el siguiente link Classes

- **fit(X,y):** Este método debe realizar el entrenamiento del modelo
- **predict(X):** Este método debe predecir la clase del input X
- **score(X,y):** Este método calcula alguna métrica de desempeño adecuada.

3. Muestre el rendimiento de su modelo para distintos dígitos, valores de *lr* y tolerancias. Recuerde separar el dataset en un conjunto de train y test.

Una forma de crear un sistema que permita clasificar dígitos en 10 clases es mediante el enfoque *One vs All*, es decir, construir 10 clasificadores binarios uno para cada dígito (un detector de 0's, un detector de 1's, etc...) con lo que la clase será escogida por el clasificador que entregue el valor de $\theta^T \phi(x)$ más grande.

1. Cree una función que implemente este sistema, puede agregar los métodos que requiera en la clase Perceptron().

2. Reporte el rendimiento de su modelo para distintos valores de *lr*, y distintas tolerancias, muestre ejemplos donde el sistema falla.

3. Explique de qué otra forma se pudo haber realizado esta tarea de clasificación multiclase utilizando el algoritmo del perceptrón binario. Explique en qué cambia el procedimiento si ahora consideramos la regresión logística para clasificación binaria.

4. Compare los resultados con lo obtenido al aplicar un algoritmo de Support Vector Machines. Para esto, estudie el algoritmo de Support Vector Classifier (SVC) implementado en `sklearn.svm`, elija un kernel que se ajuste mejor a los datos según el criterio que usted le parezca mejor, argumente su elección. Implemente el algoritmo utilizando el enfoque *One vs All*. Comente los resultados obtenidos y compare con el algoritmo implementado en la pregunta, discuta sobre los distintos enfoques que tiene y por qué un algoritmo tiene mejor rendimiento que otro.

P2. Breast Cancer Prediction

El objetivo de este problema es estudiar el comportamiento de los SVC en distintas situaciones. Para esto, se considera en esta pregunta a la base de datos de *Breast Cancer* que pueden importar con la función `load_breast_cancer` de la librería `sklearn.datasets`. Le puede ser útil utilizar la librería `pandas` para explorar los datos. La variable a predecir es *target_names*, la cual denota si un tumor es maligno o benigno. Debe considerar 3 algoritmos; Regresión Logit, SVC y Linear Discriminant Analysis (LDA), todos implementados en la librería `sklearn`. Realice lo siguiente:

1. Describa el dataset. Reporte la cantidad de columnas, datos y la cantidad de tumores malignos y benignos. Sea breve.

2. Aplique los algoritmos mencionados: Logit, SVC y LDA. Discuta sobre los resultados obtenidos. Discuta sobre los coeficientes asociados a cada variable.

3. Investigue acerca de qué representa a cada columna, elimine las columnas redundantes según cualquiera de los siguientes métodos: Elegir una columna que represente a una de las características o elegir la columna con mayor correlación con las demás y dejar esa asociada a cada característica del tumor. Escale los datos según el método para escalar que a usted le parezca mejor, argumente el por qué de su elección. Aplique los algoritmos de estudio y reporte los resultados obtenidos, discuta sobre la precisión además de el cambio en los coeficientes obtenidos en la parte anterior. ¿A qué se deben los cambios?