



## Tarea 3

En esta tarea, deberán utilizar el conocimiento sobre grafos de cómputo, para implementar un algoritmo de clasificación. En particular, deberán desarrollar una implementación parametrizable de Soft-margin SVM. Como bonus, se premiará con hasta **3.5 puntos extra** a los que implementen una ConvNet (red neuronal convolucional profunda), también utilizando grafos de cómputo parametrizables.

Para esto podrán basarse en el código disponible en el Syllabus, o también desarrollar todo desde cero. El objetivo de la tarea es que la implementación sea propia, por lo que además se prohíbe terminantemente utilizar código de otras fuentes que implementen grafos de cómputo, incluyendo bibliotecas, tareas de cursos de otras universidades, etc. Tampoco es válido citarlo e incluirlo.

### Forma funcional Soft-margin SVM

Como se mencionó anteriormente, el problema de aprendizaje a resolver corresponderá al de un Soft-margin SVM, donde será posible parametrizar algunos elementos de la estructura:

$$\begin{aligned} \operatorname{argmin}_{w,b,\{\xi_i\}} \quad & \frac{\lambda}{2} \|w\|^2 + \sum_i^N \xi_i \\ \text{s.a} \quad & y_i \cdot (w^\top x_i + b) \geq 1 - \xi_i, \forall i \in (1, N) \\ & \xi_i \geq 0, \forall i \in (1, N) \end{aligned}$$

donde  $\lambda$  es una constante que regula la importancia de ambos términos de la ecuación,  $x_i \in \mathbb{R}^K$  e  $y_i \in [-1, 1]$ . Se asume que todos los vectores son representados como vectores columna.

La parametrización de la forma funcional corresponderá al valor del hiperparámetro  $\lambda$  y la dimensionalidad  $K$  de los vectores de entrada  $x_i$ .

**Nota:** Recuerde que para poder entrenar un Soft-margin SVM mediante un grafo de cómputo, es necesario traspasar las restricciones a la función objetivo.

### Entrenamiento

Para obtener los parámetros  $(w, b)$  del Soft-margin SVM, deben utilizar descenso de gradiente estocástico sobre el grafo de cómputo, *i.e.*, tanto para obtener el valor de la función objetivo, como para calcular su derivada, el grafo de cómputo debe tener embebida la información del tamaño del mini batch.

Después de cada paso de descenso, debe imprimirse el valor de la función objetivo evaluada en el mini batch, mientras que al final de cada época de entrenamiento (un paso completo por el set de entrenamiento), se debe imprimir el valor de la función objetivo evaluada en el set de entrenamiento completo. Finalmente, el entrenamiento debe detenerse después de completar una cantidad  $N$  de épocas y se deberá imprimir en pantalla el resumen del proceso (valor de función objetivo, cantidad de descensos, cantidad de épocas, etc.) junto con los valores obtenidos para los parámetros  $w$  y  $b$ .

En relación a la parametrización, el tamaño del mini batch, el *learning rate* y la cantidad total de épocas de entrenamiento deben ser configurables.

## Formato de datos

La entrada para la tarea será un archivo de texto plano, donde la primera línea indicará la parametrización, mientras que las siguientes filas contendrán los ejemplos. La categoría de cada ejemplo se encontrará indicada en el último dato de cada fila. A continuación, un ejemplo para un Soft-margin SVM, donde los datos de entrada tienen 4 dimensiones,  $\lambda = 0.1$ , learning rate  $\alpha = 0.0001$ , mini batch de 20 ejemplos y 50 épocas de entrenamiento:

```
4,0.1,0.0001,20,50
1.5,1.1,4.6,8.3,-1
-1.9,-1.7,-1.1,-1.3,-1
8.9,-1.2,2.6,1.3,1
1.2,-8.7,3.4,5.9,1
-1.5,6.2,-2.9,7.9,1
1.4,1.9,0.5,-0.3,-1
.
.
.
```

Para ejecutar la tarea, se debe indicar por línea de comandos el nombre del archivo con los datos:

```
python tarea3.py datos.txt
```

Por favor considere que el nombre del archivo puede y va a cambiar, por lo que **no será siempre datos.txt**.

## Algunas consideraciones finales

A pesar de que la tarea no será evaluada con un gran volumen de datos, es necesario que esta tenga un tiempo de ejecución razonable (unas cuantas decenas de milisegundos por ejemplo como máximo), de manera que pueda ser corregida. Tareas que no cumplan este requerimiento, serán evaluadas con un 1.0 y deberán solicitar corrección.

## Bonus

El puntaje será asignado de acuerdo a la correctitud de la implementación de la ConvNet, además del nivel de parametrización posible, *e.g.* una red convolucional correctamente implementada, pero con estructura fija tendrá el menor puntaje de bonus (la cantidad será definida en base al conjunto de tareas entregadas). En base a esto, el formato de parametrización (primera línea del archivo) es libre para la ConvNet.

## Entrega

La tarea debe desarrollarse en Python 3.x y se recomienda utilizar NumPy para todo lo relacionado con manejo y operación de vectores y/o matrices. Su solución debe ir acompañada de un pequeño manual (escrito en un archivo de texto plano), donde se indique como ejecutarla y el formato de la salida de resultados. La entrega de la tarea tiene como fecha límite el martes 28 de noviembre a las 23:59, con un descuento por atraso de 1.0 puntos cada 6 horas o fracción, y debe realizarse en el repositorio en GitHub asignado a cada uno.

## Política de Integridad Académica

Los alumnos de la Escuela de Ingeniería deben mantener un comportamiento acorde al Código de Honor de la Universidad:

*“Como miembro de la comunidad de la Pontificia Universidad Católica de Chile me comprometo a respetar los principios y normativas que la rigen. Asimismo, prometo actuar con rectitud y honestidad en las relaciones con los demás integrantes de la comunidad y en la realización de todo trabajo, particularmente en aquellas actividades vinculadas a la docencia, el aprendizaje y la creación, difusión y transferencia del conocimiento. Además, velaré por la integridad de las personas y cuidaré los bienes de la Universidad.”*

En particular, se espera que mantengan altos estándares de honestidad académica. Cualquier acto deshonesto o fraude académico está prohibido; los alumnos que incurran en este tipo de acciones se exponen a un procedimiento sumario. Ejemplos de actos deshonestos son la copia, el uso de material o equipos no permitidos en las evaluaciones, el plagio, o la falsificación de identidad, entre otros. Específicamente, para los cursos del Departamento de Ciencia de la Computación, rige obligatoriamente la siguiente política de integridad académica en relación a copia y plagio: Todo trabajo presentado por un alumno (grupo) para los efectos de la evaluación de un curso debe ser hecho individualmente por el alumno (grupo), sin apoyo en material de terceros. Si un alumno (grupo) copia un trabajo, se le calificará con nota 1.0 en dicha evaluación y dependiendo de la gravedad de sus acciones podrá tener un 1.0 en todo ese ítem de evaluaciones o un 1.1 en el curso. Además, los antecedentes serán enviados a la Dirección de Docencia de la Escuela de Ingeniería para evaluar posteriores sanciones en conjunto con la Universidad, las que pueden incluir un procedimiento sumario. Por “copia” o “plagio” se entiende incluir en el trabajo presentado como propio, partes desarrolladas por otra persona. Está permitido usar material disponible públicamente, por ejemplo, libros o contenidos tomados de Internet, siempre y cuando se incluya la cita correspondiente.