PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
IIC-2613

TAREA 3: ALGORITMOS DE APRENDIZAJE DE MÁQUINA

LA TAREA ES INDIVIDUAL

Fecha Máxima de Entrega: 30/11, 18:00 hrs. (Siding).

1 Objetivo

Muchach@s, en esta tarea tendrán la oportunidad de poner en práctica los conocimientos que han aprendido en clases. En particular, probarán el rendimiento de diferentes clasificadores para resolver las denominadas bAbI-tasks https://research.fb.com/downloads/babi/.

2 bAbI-tasks Dataset

El set de datos que deberán usar para esta tarea son las bAbI tasks. Las bAbI tasks fueron publicadas el año 2015 con el fin de evaluar la capacidad de comprensión de lectura de modelos de *machine learning* mediante tareas de preguntas y respuestas.

Los autores del set de datos generaron un total de 20 tareas y cada una evalúa alguna capacidad de razonamiento diferente. Cada tarea cuenta con su set de entrenamiento y test respectivo. A modo de ejemplo, a continuación se muestran instancias del set de entrenamiento para la tarea número 1:

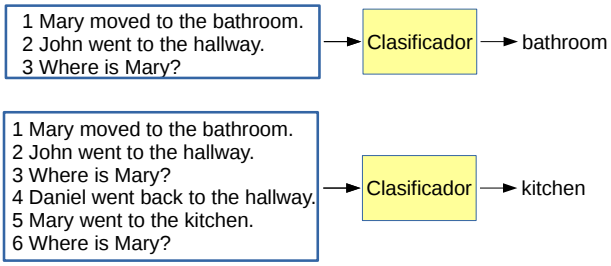
```
1 Mary moved to the bathroom.
2 John went to the hallway.
3 Where is Mary?        bathroom        1
4 Daniel went back to the hallway.
5 Sandra moved to the garden.
6 Where is Daniel?      hallway 4
7 John moved to the office.
8 Sandra journeyed to the bathroom.
9 Where is Daniel?      hallway 4
10 Mary moved to the hallway.
11 Daniel travelled to the office.
12 Where is Daniel?     office  11
13 John went back to the garden.
14 John moved to the bedroom.
15 Where is Sandra?     bathroom        8
```

Cada línea de texto comienza con un identificador correlativo. Cuando el identificador se reinicia al valor 1 indica el inicio de una nueva historia. Luego del identificador, cada línea puede contener 2 posibles fuentes de información: i) Un hecho declarando algún conocimiento, por ejemplo, ”Mary moved to the bathroom”, o ii) Una secuencia que contiene una pregunta (”Where is Mary?”) seguida de una repuesta (”bathroom”) y luego un número indicando el identificador del hecho que justifica la respuesta (supporting fact, ej: ”1”).

Todas las tareas del set de datos bAbI tasks siguen la misma estructura. En cada caso, el objectivo es utilizar los hechos de cada historia para predecir una correcta respuesta a cada pregunta. Acompañando a este enunciado, se subirán a SIDING el set de datos y una presentación con una breve descripción de cada una de las tareas.

3 Preprocesamiento de los datos

Para entrenar y probar sus clasificadores deberán procesar línea a línea los sets de entrenamiento y test de cada una de las tareas del set bAbI-tasks. Específicamente, para cada tarea deberán ir formando los registros que posteriormente usarán para entrenar sus modelos. Se sugiere que los registros sean tuplas de la forma `(story, question, answer)`. Es importante mencionar que `story` es la historia, o secuencia de hechos relevantes para contestar cada pregunta, es decir, **no contiene los hechos posteriores a la pregunta**. A modo de ejemplo, considere el siguiente caso donde la respuesta a la misma consulta cambia de acuerdo a la historia considerada:



Adicionalmente, durante la creación de tuplas debe llevar a cabo el proceso de *tokenización*. El proceso de *tokenización* consiste en aplicar una función que recibe un *string* y retorna una lista de *tokens*. Se espera que la función *tokenizadora* que usen transforme a minúsculas el texto y remueva la puntuación. A continuación un ejemplo de la transformación esperada: `tokenize('Mary moved to the bathroom.') == ['mary', 'moved', 'to', 'the', 'bathroom']`.

Considerando lo anterior, a continuación se muestra la primera tupla `(story, question, answer)` generada luego de procesar el ejemplo del set de entrenamiento mostrado en la sección 2:

```
(
    [
        ['mary', 'moved', 'to', 'the', 'bathroom'],
        ['john', 'went', 'to', 'the', 'hallway']
    ],
    ['where', 'is', 'mary', '?'],
    'bathroom'
)
```

En forma opcional, en el desarrollo de su tarea puede aplicar los siguientes pasos adicionales de preprocesamiento a los textos del set de datos bAbI-tasks.

- *Eliminación de stopwords*: esta etapa de preprocesamiento consiste en eliminar del set de datos una lista de palabras no informativas, como artículos y ciertas preposiciones, las cuales no aportan a la clasificación.
- *Aplicación de stemming*: esta etapa de preprocesamiento consiste en transformar sustantivos y verbos a su forma base o raíz. Por ejemplo, transformar palabras como *cats* a *cat*, o *running* a *run*. Como veremos mas adelante, en la tarea se recomienda usar la herramienta de stemming propuesta por Martin Porter.

Para realizar estos procesos, se recomienda usar la librería `nltk`. Ejemplos de cómo realizar este procesos se muestran al final de este documento.

4 Espacio de características

Para aplicar técnicas de aprendizaje de máquina, la primera tarea será llevar a un espacio de características (feature space) cada uno de los textos que describen los ejemplos de entrenamiento. En particular, en esta

tarea llevaremos los textos a un espacio de características usando la codificación one-hot. En el contexto de procesamiento de texto, la codificación one-hot consiste en representar cada palabra con un vector de dimensionalidad igual al vocabulario del set de datos correspondiente. Donde vocabulario se refiere a la lista de palabras distintas que componen el set de datos. Así, la codificación one-hot consiste en representar la palabra i-ésima del vocabulario mediante un vector en que todas sus componentes son 0, a excepción de la componene i-ésima, la cual tiene un valor 1. A modo de ejemplo, la codificación one-hot para el vocabulario *V=*$\{$*la, le, li, lo, lu*$\}$ es *OneHot(V):*$\{$*la=00001, le=00010, li=00100, lo=01000, lu=10000*$\}$.

Para el desarrollo de esta tarea deberán encontrar dos vocabularios:

- *tokens en hechos y preguntas*: correspondiente a todos los posibles tokens presentes en hechos y preguntas.
- *tokens en respuestas*: correspondiente a todas las posibles respuestas.

Para la formación de los vocabularios deben considerar la totalidad de las historias del set de entrenamiento y test al mismo tiempo. Estos vocabularios serán los que deben usar para construir las representaciones one-hot de los tokens.

Si bien la codificación one-hot permite llevar a un espacio de características cada palabra de un vocabulario, la pregunta que surge es: ¿cómo representar cada oración?. Una forma simple de hacer esto es mediante el método *bag–of–words* que consiste en sumar las representaciones de las palabras pertenecientes a la oración. Este método será usado para construir las representaciones de los hechos y consulta.

El siguiente paso es unir en un ejemplo de entrenamiento las distintas oraciones de cada historia. Para esto seguirán la estrategia de concatenar las vectores que representan los hechos correspondientes. Adicionalmente a los hechos, también es necesario concatenar la consulta correspondiente. Para ejemplificar, si el tamaño del vocabulario de una tarea es n y existe un registro que cuenta con m hechos, el vector que entrará a los modelos será de dimensión $n * (m + 1)$. El término $m + 1$ toma esa forma porque considera todos los hechos además de la consulta.

Una consideración importante es que debido a que algunas historias tienen distinto número de hechos y que para cada problema los clasificadores esperan una dimensionalidad fija de entrada, será necesario rellenar con ceros las entradas cuyo largo de historia no sea el máximo de la tarea correspondiente.

Finalmente, para representar las posibles respuestas, deberán usar la estrategia one-hot utilizando como vocabulario el de todas las posibles respuestas.

5 Actividades

5.1 Máquinas de Vectores de Soporte

- Utilice el set de datos bAbI-tasks para entrenar un clasificador del tipo SVM para cada tarea del set bAbI-tasks, según las siguientes indicaciones:
  - Utilice un kernel lineal.
  - Penalidad del tipo "l_2" para los pesos.
  - Penalidad el tipo "squared_hinge" para las variables slack.
- Pruebe distintos valores para el coeficiente de penalización de las variables slack. Comente sus resultados.
- Pruebe el rendimiento del clasificador en el set de entrenamiento. Nota diferencias respecto del resultado en el set de test, comente.
- ¿Qué categoría obtiene el mejor rendimiento?, ¿Cuál es el más bajo?. Indique razones que puedan justificar estos resultados.

- Finalmente, pruebe el rendimiento de un SVM que se entrene utilizando todas las tareas del set bAbI-tasks. ¿Observa diferencias en el tiempo de ejecución y exactitud del modelo resultante?. En general, ¿ cómo se compara el modelo conjunto con el rendimiento de cada uno de los clasificadores individuales entrenados en cada problema del set bAbI-task?

5.2 Redes Neuronales

- Utilice el set de datos bAbI-tasks para entrenar un clasificador del tipo red neuronal para cada tarea del set bAbI-tasks, según las siguientes indicaciones:
  - Para la capa de entrada utilice tantas neuronas como atributos.
  - Para la capa de salida utilice codificación one-hot.
  - Para la capa oculta ajuste empíricamente el número de neuronas.
  - Entrene las redes utilizando gradientes estocástico (método incremental visto en clases) o alguna de sus variantes disponible en la librería de software utilizada.
- ¿Cómo determinó el número de neuronas de la capa oculta?
- ¿Qué criterio utilizó para terminar de iterar el proceso de entrenamiento?.
- ¿Cuál es la exactitud promedio en cada set, entrenamiento y test?.
- ¿Observa sobreajuste?, comente.
- ¿Qué tarea alcanza el mejor rendimiento?, ¿Cuál es el más bajo?. Indique razones que puedan justificar estos resultados.
- Finalmente, pruebe el rendimiento de una red neuronal que se entrene utilizando todas las tareas del set bAbI-tasks. ¿Observa diferencias en el tiempo de ejecución y exactitud del modelo resultante?. En general, ¿cómo se compara el modelo conjunto con el rendimiento de cada uno de los clasificadores individuales entrenados en cada problema del set bAbI-tasks?

5.3 Resultados Comparativos

Compare su experiencia utilizando las técnicas de SVM y Redes Neuronales. En su análisis considere lo siguiente:

- Facilidad de uso.
- Tiempo de proceso.
- Exactitud de la clasificación.
- Tipos de error más frecuentes.

5.4 Formato de Entrega

La tarea debe ser desarrollada en un Jupyter Notebook https://jupyter.org/ corriendo sobre un kernel de Python 3. Para entregar la tarea deben entregar el *notebook* exportado en formato HTML, para respaldar los *outputs*, y en formato .ipynb, para poder replicar los experimentos.

5.5 Herramientas de Software

Para el entrenamiento y prueba de los clasificadores use la librería scikit-learn de Python, disponible en: http:\www.scikit-learn.org. En clases y ayudantía de la tarea se explicará el funcionamiento de esta librería y se mostrarán algunos ejemplos.

Para agregar una etapa de procesamiento de los textos puede utilizar la librería de text-mining nltk: http://www.nltk.org, a modo de ejemplo considere el sigueinte código que implementa una etapa de eliminación de stopwords y aplicación de stemming.

```
## Loading Stopwords model
from nltk.corpus import stopwords

# Loading Stemmer model
from nltk.stem import PorterStemmer
stemmer = PorterStemmer() # stemmer
st = stemmer.stem  # get the stemming function
```