Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

IIC2613 – Inteligencia Artificial (I/2019)

Tarea 2 compresión de texto usando redes neuronales profundas

En esta tarea, deberán extender el sistema de compresión de la tarea 1, utilizando redes neuronales profundas, en vez de los algoritmos vistos previamente en clases. El uso de redes profundas permitirá, en teoría, mejorar fuertemente los resultados de compresión obtenidos en la tarea 1.

Codificación basada en frecuencia

Los esquemas tradicionales de codificación de texto, utilizan una cantidad fija de bits para representar todos los caracteres. Por ejemplo, en el caso de la codificación EASCII, se utiliza 1 byte (8 bits) para representar cualquiera de los 256 caracteres que soporta. A pesar de que esta simplicidad trae grandes beneficios, los archivos de texto que obedecen a este estilo de codificación tienden a ser más pesados de lo necesario, debido a que contienen una gran cantidad de redundancia en la codificación.

Una forma de remediar esto, es utilizar esquemas de codificación basados en frecuencia. Estos se basan en el hecho de que de manera natural, algunas letras, sílabas y/o palabras ocurren con mayor frecuencia que otras en los lenguajes humanos. De esta manera, los caracteres con más probabilidad de aparecer reciben una codificación que requiere menos bits que las codificaciones de aquellos caracteres que ocurren menos frecuentemente.

Existe variados algoritmos para realizar codificación basada en frecuencia. Dos de los más utilizados son la **codificación de Huffman**, o la **codificación aritmética**.

Codificación adaptativa basada en contexto

Siguiendo la misma lógica de la codificación basada en frecuencia, es posible extender estos esquemas al tomar en consideración el hecho que, dependiendo del contexto, hay algunos caracteres, palabras y/o sílabas que es más probable que ocurran que otras. Por ejemplo, si desde un archivo de texto se lee "*inteligencia arti*", es altamente probable que el siguiente carácter a leer desde el archivo, sea la letra "*f*". Así, un algoritmo de codificación adaptativa usaría este conocimiento para utilizar le menor cantidad de bits posibles para codificar los caracteres más probables, y en particular para este caso, la letra "*f*".

Tomando todo lo anterior en consideración, en esta tarea deberá implementar un esquema de codificación (y decodificación) de texto adaptativo, de tal manera que sea capaz de comprimir archivos de texto simple, y luego sea capaz de descomprimirlos, sin que exista pérdida de información. El esquema debe utilizar una red neuronal profunda para realizar las predicciones en base al contexto, que deben ser entregadas a un codificador basado en frecuencia en forma de una tabla de frecuencia/probabilidades.

Set de datos

Para entrenar sus modelos, puede usar cualquier set de datos de texto disponible, tales como **20 Newsgroups**, o el **Reuters News dataset**. Se recomienda que el set de datos sea lo más extenso posible, con el fin de capturar la mayor cantidad de patrones.

Modelos de aprendizaje supervisado

En la tarea puede utilizar cualquier arquitectura basada en redes neuronales profundas. No se permite el uso de modelos previamente entrenados. La red debe ser implementada utilizando Pytorch 1.0 o mayor, para lo cual puede instalar los módulos de manera local, o utilizando Google Colaboratory.

Entrega y evaluación

La tarea debe ser realizada en Python 3.5 o superior. Su entrega debe incluir, al menos, dos archivos. El primero, llamado `tarea2.py`, debe implementar el compresor/descompresor de archivos de texto. El segundo archivo, llamado `tarea2_train.ipynb`, debe contener un Jupyter Notebook donde se implemente y describa el proceso de entrenamiento de la red y la construcción del sistema. Este archivo podrá ser ejecutado eventualmente al momento de la corrección, por lo que se recomienda incluir todo lo necesario.

Con respecto a la ejecución del compresor/descompresor, este debe permitir su ejecución a través de la línea de comandos, utilizando parámetros para especificar su modo de ejecución. Específicamente, el sistema deber ser llamado utilizando el siguiente esquema:

- `python tarea2 -c <src> <dst>`: comprime el archivo de texto de nombre `<src>`, generando el archivo comprimido de nombre `<dst>`.
- `python tarea2 -d <src> <dst>`: descomprime el archivo comprimido de nombre `<src>`, generando el archivo de texto de nombre `<dst>`.

Todas las tareas serán evaluadas utilizando el mismo conjunto de archivos de prueba, todos codificados en EASCII. Existirán bonos para las tres tareas que presenten mejor razón de compresión en los archivos de prueba.

La entrega de la tarea tiene como fecha límite el viernes 17 de mayo a las 23:59, y debe realizarse en la carpeta `T2` del repositorio en GitHub asignado a cada uno. Para fines de corrección, se revisará la última versión subida al repositorio. Finalmente, se aplicará un descuento de 1.0 ptos. cada 6 horas o fracción de atraso.

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.