

Proyecto PAPIME 2018: PE103018

Manual de prácticas de laboratorio de la asignatura Redes Neuronales

Plan de trabajo / Prácticas a elaborar

1. Instalación de requerimientos
 - Kubuntu 18.04, Python 3, pip3, numpy, scipy, matplotlib, scikit-learn, pandas, jupyter, cuDNN, tensorflow-gpu, keras y opencv.
2. Implementación del algoritmo de propagación del error hacia atrás con NumPy (*Forward Step*)
 - Para una red neuronal de 3 capas.
3. Implementación del algoritmo de propagación del error hacia atrás con NumPy (*Backwards Step*)
 - Para una red neuronal de 3 capas.
4. Implementación del algoritmo de propagación del error hacia atrás con NumPy (*Forward Step*)
 - Para una red neuronal de N capas.
5. Implementación del algoritmo de propagación del error hacia atrás con NumPy (*Backwards Step*)
 - Para una red neuronal de N capas.

Elabora: Sergio

6. Introducción a la implementación de redes neuronales con la librería Keras
 - Clasificación de señales de sonar
 - Análisis de regresión para determinar el precio de venta de una casa
 - Validación cruzada
7. Determinación automática de los metaparámetros de la arquitectura de la red neuronal
 - *Checkpointing*
 - Búsqueda en *mall*a (incluyendo validación cruzada) de los metaparámetros de la red neuronal (*GridSearchCV*)
 - Búsqueda *aleatoria* (incluyendo validación cruzada) de los metaparámetros de la red neuronal (*RandomSearchCV*)
8. Implementación de redes neuronales convolucionales (*CONVNETS*) con Keras
 - Clasificación de dígitos numéricos manuscritos (conjunto de datos -imágenes- *MNIST*)
 - Determinación de accidentes -daños- en la retina mediante fotografías retinales (tesis de Carlos Montiel)
 - Score F_1

Elaboran: Alejandra, Karen y Nancy

9. Detección de sismos en señales de sismógrafo registradas en Ometepepec, Guerrero

- Con CONVNETS
- Con redes neuronales recurrentes
 - Con unidades tipo LSTM
 - Con unidades tipo GRU

Elaboran: Gerardo (interpretación de señales sismográficas), Armando, Fernando, Hugo y Rodrigo

10. Aprendizaje por refuerzo aplicado a la resolución del ajedrez

Elabora: Héctor (de la Rosa)

11. Todas las que propongan (¿Nancy y Alejandra sobre neurociencia?, ¿Gerardo con algo más sobre sismología?, ¿Hugo, Fernando, Armando y/o Rodrigo?, ¿Karen sobre algo de lo que hizo en Inglaterra -aplicará-?, ...)

12. Propuestas en el *aire*: Optimizadores, transferencia de estilo, predicción de series de tiempo caóticas con ruido (trabajo de la UMSNH: José Rafael y el Dr. Flores), GANs, máquinas (restringidas, tal vez) de Boltzmann, YOLO, ...

Formato de práctica

- Látex... Por favor busquen en Internet una plantilla que les agrade para que todos manejemos el mismo diseño.
 - Secciones o apartados (algunas las pueden omitir si así lo consideran conveniente):
 - Número y nombre
 - Objetivo
 - Requerimientos: Paquetes de Python 3, conjuntos de datos (como el MNIST o el de Ometepepec),...
 - Introducción / antecedentes
 - Desarrollo
 - Resultados y/o conclusiones
 - Ejercicios adicionales
 - Referencias: Bibliografía en la que se basaron, URLs, fuentes de los conjuntos de datos y de las imágenes que empleen, etc... Por favor, ¡no pongan Wikipedia, Rincón del Vago, etc.! ... Gracias
-

Recursos

Bibliografía e imágenes

- Libros **Deep Learning: From Basics to Practice, Volume 1 & 2**: el volumen 1 se dedica más a los fundamentos del aprendizaje de máquina y el 2 al *deep learning*. Estos libros se los haré llegar por correo, son bastante interesantes porque manejan la parte matemática a un nivel de sumas y multiplicaciones solamente. ¡No son gratuitos! (por favor, no los compartan indiscriminadamente :) ... y sí los voy a comprar, al menos un tomo de cada uno -aunque sólo existen en formato digital-).
- <https://dlbasics.com/>: Página web de los libros anteriores, en ella pueden encontrar todas las imágenes de ambos volúmenes para utilizarlas en las prácticas que elaboren. Las imágenes son de muy buena calidad y excelentes para ilustrar los conceptos de las redes neuronales y el aprendizaje profundo, aparte de que el autor permite su uso de forma gratuita (referenciándolo a él como autor, solamente) y de que proporciona las libretas de Jupyter con las que se crearon, para quien guste generarlas por sí mismo.

Cómputo

- **Servidor de Raggi** (IP: 10.99.1.72 y GPU con 2,048 CUDA cores): Pueden acceder usando el shell seguro (ssh), aunque sólo desde la ENES, empleando las cuentas siguientes:
 - Gerardo: grodriguez
 - Nancy: norigel
 - Karen: kreyes
 - Alejandra: aesquivel
 - Rodrigo: rcortez
 - Hugo: hgodinez
 - Armando: jortiz
 - Fernando: faguilar
 - Todos con contraseña: **linux** (cámbienla a algo más robusto por favor, si no, Raggi me ahorca)
- **PC de escritorio de Sergio** (IP: 10.99.1.60 y GPU con 640 CUDA cores): No es tan potente como el servidor de Raggi, pero sí nos puede ayudar. También pueden acceder con ssh y las mismas cuentas (también cambien la contraseña, por favor -OJO: son 2 computadoras diferentes, por tanto, son 2 cuentas de usuario diferentes).

-
- Nancy, Karen y Ale: En la cuenta de Nancy, en ambas computadoras, puse el conjunto de datos de Carlos Montiel (las imágenes retinales para el entrenamiento de las CONVNETS, bueno, las zonas dañadas extraídas y los ejemplos sin daño).

Tamaño	Archivo
43 MB	dataset_t.zip

- Descompactado el archivo la estructura de directorios queda (el prefijo “mas_” se refiere a daño retinal, pero no recuerdo el término técnico):

Archivos	Directorio
----------	------------

Archivos	Directorio
83,194	dataset_train/
75,257	dataset_train/mas_negativos/
7,937	dataset_train/mas_positivos/

Archivos	Directorio
9,653	dataset_test/
9,011	dataset_test/mas_negativos/
642	dataset_test/mas_positivos/

- OJO: Son muchos archivos del conjunto *train*, por eso no los descompacté. Además, las imágenes NO tienen el mismo tamaño (en ambos conjuntos), por eso deben preprocesarlas escalándolas al tamaño de la imagen más grande (según yo, lo pueden hacer con funciones de OpenCV o de SciPy).

-
- Gerardo, Armando, Fernando, Hugo y Rodrigo: En la cuenta de Gerardo, en ambas computadoras, puse los archivos *.sac* que Luis Antonio me pasó de las señales sismológicas registradas en Ometepec, Guerrero.

Tamaño	Directorio	Nº de archivos .sac
838 MB	Ometepec	53,358

- OJO: Tengan cuidado, comandos como “ls Ometepetec/*.sac” no funcionan, porque Linux intenta substituir “*.sac” por los 53,358 nombres reales, lo que supera el límite máximo de 250 (+/-) caracteres para una sola línea de Bash.