

Proyecto de curso

Curso: Técnicas Avanzadas de Data Mining y Sistemas Inteligentes (INF659)
Docentes: Dr. César Beltrán Castañón
 Mag. César Olivares Poggi
 Mag. Renato Hermoza Aragonés

El proyecto de curso permitirá que los estudiantes de postgrado apliquen su experiencia en el diseño, implementación y evaluación de algoritmos de aprendizaje profundo. Para su elaboración se conformará grupos de hasta cuatro alumnos cada uno. El objetivo principal de la actividad será la experimentación con metodologías del estado del arte en aprendizaje profundo aplicando buenas prácticas de investigación. Se requerirá reportar y reproducir a manera de línea base la experimentación con un modelo relevante del estado del arte, y luego realizar una aplicación propia a un problema de su elección aplicando las mejores prácticas en investigación.

El proyecto tendrá un peso del 50% de la nota final y constará de los siguientes entregables:

Actividad/Entregable	Fecha límite de entrega	Puntaje
Elección de tema y conformación de grupos	21 de agosto	–
1. Jupyter Notebook con ejecución comentada del modelo base	1 de octubre	3 puntos
2. Diapositivas y presentación en clase del modelo base y de la propuesta de proyecto de aplicación	9 de octubre	4 puntos
3. Primera parte del informe escrito (Introducción, estado del arte y metodología), en formato IEEE, tamaño A4, máximo 3 páginas de extensión	29 de octubre	3 puntos
4. Jupyter Notebook con ejecución comentada del proyecto de aplicación	12 de noviembre	3 puntos
5. Entrega de poster	19 de noviembre	1 puntos
6. Presentación en Workshop abierto.	20 de noviembre	3 puntos
7. Informe final, en formato IEEE, tamaño A4, máximo 8 páginas de extensión	26 de noviembre	3 puntos

Elección del tema

Cada grupo deberá elegir uno de los temas que se indica a continuación. Para cada uno de ellos se indica la referencia de un modelo de Deep Learning que puede ser tomado como línea base, así como el enlace a una implementación en TensorFlow disponible.

Si un grupo lo desea, dentro del mismo tema asignado, puede elegir tomar como base otro modelo de Deep Learning y/o otra implementación en TensorFlow del modelo elegido. El cambio de modelo base puede realizarse en coordinación con el profesor a más tardar el día 6 de setiembre.

Cada tema podrá ser elegido por un máximo de 2 grupos, con tal que no se tome como base el mismo modelo de Deep Learning.

1. **Clasificación de imágenes**

Modelo base: ResNet-50

Kaiming H., Xiangyu Z., Shaoqing R., Jian S. (2015). Deep Residual Learning for Image Recognition

<https://github.com/tensorflow/models/tree/master/official/resnet>

2. **Segmentación de imágenes**

Modelo base: U-Net

Olaf R., Philipp F., Thomas B. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation.

https://github.com/jakeret/tf_unet

3. **Anotación de imágenes**

Modelo base: Show and Tell

Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2017). Show and Tell: Lessons Learned from the 2015 MSCOCO Image Captioning Challenge. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(4), 652–663.

<https://github.com/tensorflow/models/tree/master/im2txt>

4. **Transferencia de estilos**

Leon A., Alexander S., Matthias B. (2015). A Neural Algorithm of Artistic Style.

<https://github.com/anishathalye/neural-style>

5. **Redes generativas adversarias**

Modelo base: GAN (Vanilla)

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... Bengio, Y. (2014). Generative adversarial nets. In Advances in neural information processing systems (pp. 2672–2680).

<http://wiseodd.github.io/techblog/2016/09/17/gan-tensorflow/>

6. **Reconocimiento y generación de audio**

Modelo base: DeepMind WaveNet

Oord, A. van den, Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., ... Kavukcuoglu, K. (2016). WaveNet: A Generative Model for Raw Audio.

<https://github.com/buriburisuri/speech-to-text-wavenet>

7. **Representación vectorial de palabras**

Modelo base: Word2Vec

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In NIPS (pp. 1–9).

<https://github.com/tensorflow/models/blob/master/tutorials/embedding/word2vec.py>

8. **Traducción automática**

Modelo base: Seq2Seq

Vinyals, O., Kaiser, L., Koo, T., Petrov, S., Sutskever, I., & Hinton, G. (2014). Grammar as a Foreign Language. arXiv:1412.7449

<https://github.com/tensorflow/models/tree/master/tutorials/rnn/translate>

Ejecución de modelo base

El primer entregable consistirá en la presentación de un Jupyter Notebook en el que se muestre haber ejecutado satisfactoriamente la implementación del modelo base en TensorFlow:

- **Explicación del modelo.** El notebook debe exponer por separado, comentando en celdas de *markdown*, las principales funciones y secciones del código de implementación.
- **Entrenamiento.** Se debe mostrar el entrenamiento del modelo, siempre que se encuentre disponible, con al menos uno de los conjuntos de datos con los que fue experimentado en el artículo de referencia. Por motivos de capacidad computacional puede ser suficiente mostrar el entrenamiento del modelo por un número reducido de épocas.
- **Evaluación del aprendizaje.** Se debe mostrar métricas y una gráfica de la curva de aprendizaje del modelo tanto en el conjunto de entrenamiento como en el conjunto de validación. Por motivos de eficiencia computacional, las mediciones en el conjunto de validación pueden tomarse con menor granularidad que en el conjunto de entrenamiento.
- **Carga de pesos del modelo entrenado.** Diversas implementaciones de referencia en TensorFlow vienen acompañadas de pesos pre-entrenados o *checkpoint*. Se debe mostrar haber cargado satisfactoriamente dichos pesos o *checkpoint* y haber utilizado el modelo pre-entrenado con muestras de ejemplo.

Presentación en clase del modelo base y propuesta de proyecto

El segundo entregable corresponde a la presentación en clase del avance realizado y la propuesta de proyecto de aplicación. Se entregará las diapositivas a emplearse en la presentación y una propuesta de proyecto para la segunda mitad del curso. La presentación tendrá una duración máxima de 25 minutos, distribuidos de la siguiente manera:

- Presentación del modelo base (Diapositivas PowerPoint o equivalente, 8 minutos)
 - o Contexto y motivación del modelo base
 - o Arquitectura del modelo base
 - o Experimentación y resultados reportados en la publicación de referencia
- Implementación en TensorFlow (Jupyter Notebook, 7 minutos)
 - o Presentación de la ejecución del modelo base (no hace falta preparar nuevas diapositivas, se puede emplear el primer entregable)
- Propuesta de proyecto (Diapositivas PowerPoint, 3 minutos, ver abajo)
- Preguntas y respuestas (7 minutos)

Propuesta de proyecto

La propuesta de proyecto definirá el trabajo a ser realizado en la segunda parte del curso. Este trabajo consistirá en la aplicación del modelo base o una variación de éste a un problema de su elección aplicando las mejores prácticas en investigación. La propuesta de proyecto deberá incluir lo siguiente:

- Nombre del proyecto
- Objetivo del proyecto
- Conjunto(s) de datos a utilizar
- Artículos científicos relevantes
 - o Referencia de al menos 4 artículos relevantes por su semejanza al problema a abordar.

Informe del proyecto

El informe debe incluir la siguiente información:

- **Introducción**

- o Presentación del problema general sobre el que versará el trabajo, su dificultad y/o relevancia, y cómo se integra dentro del campo del aprendizaje automático
 - o Objetivo del estudio
 - o Organización del informe (secciones)
- **Estado del arte**
 - o Breve síntesis del aporte que otros artículos científicos han realizado para este problema: cómo se ha abordado anteriormente este problema (antes y después de Deep learning), cuáles son las publicaciones con mejores resultados hasta la fecha y qué caracteriza a sus propuestas. Si existe una revisión sistemática o *survey* sobre el tema, indicarlo.
- **Metodología**
 - o De ser el caso, qué se requiere en cuanto preprocesamiento de datos, selección y/o extracción de características.
 - o Selección y justificación de la medida de calidad que se usará para evaluar el rendimiento del modelo.
 - o Explicación y diagrama de la arquitectura del modelo que se propone.
 - o Estrategia de validación a emplear para el ajuste de hiperparámetros (p.ej., búsqueda en grilla, búsqueda aleatoria, etc.).
 - o Explicación y diagrama del proceso propuesto, desde la adquisición de los datos hasta la evaluación del rendimiento. Asegurarse de nunca emplear el conjunto de pruebas en tareas de entrenamiento, validación ni ajuste de hiperparámetros.
- **Experimentación y resultados**
 - o Descripción del conjunto de datos
 - Origen del conjunto de datos
 - Según corresponda, número y tipo de características, dimensiones, resolución, profundidad, etc.
 - Número de muestras en los conjuntos de entrenamiento, validación y prueba. En caso aplique, número de muestras por clase.
 - o Descripción del entorno de experimentación empleado (hardware y software).
 - o Línea base: Reproducción de resultados reportados en una publicación anterior.
 - o Reporte del entrenamiento del modelo con diversos hiperparámetros. Incluir curvas de aprendizaje de la función de pérdida en el conjunto de entrenamiento y de validación. Indicar configuración de los modelos, número de épocas de entrenamiento, tiempo de ejecución y evaluación del rendimiento usando la medida de calidad elegida.
- **Discusión**
 - o Comparación de línea base y resultados propios.
 - o Interpretación de los resultados obtenidos.
 - o Identificación y visualización de ejemplos en los que tienen dificultad los modelos ensayados. ¿A qué se podría atribuir?
 - o ¿Cómo podría ser mejorado su sistema?
- **Conclusiones y trabajos futuros**

Código y/o scripts (Jupyter Notebooks)

- El código de implementación deberá presentarse en Jupyter Notebooks, empleando Python con TensorFlow.
- Se recomienda usar GitHub como entorno de colaboración.
- Si el equipo desea trabajar en otro entorno, o en un lenguaje de programación diferente, deberá consultarlo previamente con el profesor.

- Se deberá asignar nombres representativos a los archivos, de manera que se pueda identificar su orden relativo y el propósito de cada uno.
- No se aceptará código simplemente copiado de otras fuentes.
- En caso de adaptar código existente, citar siempre la fuente.
- El código deberá estar mínimamente comentado.
- No se aceptará comentarios en otros idiomas.

Recursos

- Buscador de literatura académica
<https://scholar.google.com.pe/>
- Acceso a bases de datos electrónicas – PUCP
<http://biblioteca.pucp.edu.pe/recursos-electronicos/bases-de-datos/>
- Formato IEEE (MS Word y LaTeX)
https://www.ieee.org/conferences_events/conferences/publishing/templates.html
- Editor colaborativo LaTeX en línea
<https://www.overleaf.com/>
- Working with Jupyter Notebook files on GitHub
<https://help.github.com/articles/working-with-jupyter-notebook-files-on-github/>
- Algunas guías para elaborar un póster:
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1876493/>
<https://nau.edu/undergraduate-research/poster-presentation-tips/>
<http://guides.nyu.edu/posters>