

EVALUACIÓN PARCIAL N°2



NOMBRE DE LA EVALUACIÓN: Conociendo los Fundamentos de Deep Learning

INFORMACIÓN GENERAL:

| Sigla | Nombre Asignatura | Tiempo Asignado | % Ponderación |
|---------|-------------------|-----------------|---------------|
| DLY0100 | Deep Learning | 180 minutos | 40% |

AGENTE EVALUATIVO:



Heteroevaluación (docente)

TABLA DE ESPECIFICACIONES

| RESULTADOS DE APRENDIZAJE | INDICADORES DE LOGRO (IL) | % PONDERACIÓN IL |
|---|---|------------------------|
| RA1 Programa un modelo de Deep Learning de clasificación de objetos utilizando técnicas de aprendizaje supervisados para resolver un caso de estudio. | IL1.1 Ejecuta los procedimientos fundamentales de carga de datos, prueba de funciones de activación y entramiento de la red neuronal aplicando los conceptos fundamentales de Deep Learning, tales como Red Neuronal Artificial, Perceptrón y Perceptrón multicapa. | 20% |
| | IL1.2 Selecciona funciones de activación, error y salida, aplicada a redes neuronales artificiales simples, de acuerdo con el caso de estudio planteado utilizando técnicas fundamentales de DL. | 30% |
| | IL1.3 Programa algoritmos en lenguaje Python, para optimizar un modelo de red neuronal artificial | 30% |

| | | |
|--------------|---|-------------|
| | (MLP), utilizando Tensorflow mediante Keras en el contexto de un caso de estudio. | |
| | IL1.4 Evalúa el desempeño de los modelos utilizando métricas de acuerdo con el caso de estudio presentado. | 20% |
| Total | | 100% |

INSTRUCCIONES

Indicaciones para la entrega de encargo con presentación.

- Esta evaluación se debe realizar en grupos de dos integrantes y tiene un 40% de ponderación sobre la nota final de la asignatura.
- El tiempo para desarrollar esta evaluación es de 180 minutos y se desarrollará en el taller de Alto Computo, Laboratorio de Datos.
- Los grupos deberán elaborar el encargo con presentación que constituye el primer hito en el desarrollo del examen transversal de la asignatura.
- El encargo con presentación deberá comunicar el proceso de elaboración de una red neuronal MLP, que es capaz de resolver un problema de manera óptima basándose en el caso de uso planteado para el caso semestral.
- Aplica la metodología CRISP.DM, considerando cada una de las fases.
- Para la presentación de los grupos deberán exponer en un tiempo de cinco a ocho minutos, adicionalmente se asignarán cinco minutos para realizar preguntas a la dupla (en esta evaluación la dimensión individual será formativa).
- La presentación deberá ser en base a una PPT donde explica el proceso de desarrollo.
- Deberá responder a las preguntas planteadas, plasmando las respuestas en la PPT de presentación.

Contexto del caso Forma B.

A continuación, se presenta el caso a ser utilizado en las tres evaluaciones sumativas y el examen transversal de la asignatura. Deberá ser capaz de implementar un proyecto de Deep Learning entrenando un modelo ajustado a partir de los datos entregados para este caso. Desarrollará el proyecto utilizando las etapas de metodología CRISP.DM, para la toma de las mejores decisiones de la ejecución del proyecto.

Para este caso un Banco requiere la digitalización de cheques utilizando técnicas de Deep Learning.

Para contextualizar el caso, el problema del reconocimiento de texto manuscrito en imágenes es muy importante en cuanto a aplicaciones en el área de visión por computadora. Un problema más acotado, pero no menos importante, es el de reconocimiento de secuencias de dígitos manuscritos de largo variable en imágenes. El presente trabajo consiste en investigar un nuevo enfoque para resolver el problema de reconocimiento de secuencias de dígitos manuscritos en imágenes. Se plantea dicho problema como uno de detección de objetos en que los dígitos corresponden a los objetos a detectar.

Para el entrenamiento se cuenta con un dataset llamado MNIST (el cual cuenta con 60.000 ejemplos para entrenar la red y otros 10.000 ejemplos para testear en modelo creado. Los dígitos se normalizaron en tamaño y se centraron en una imagen de tamaño fijo.

Descargar dataset a utilizar según el caso práctico entregado en el enunciado, o bien desde <http://yann.lecun.com/exdb/mnist/> o <https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz>

Pauta de Evaluación

RÚBRICA EVALUACIÓN 1 PARTE GRUPAL

Tabla 1: Descripción por categoría de respuesta

| Categoría | % logro | Descripción |
|-----------------------------|---------|--|
| Muy buen desempeño | 100% | Demuestra un desempeño destacado, evidenciando el logro de todos los aspectos evaluados en el indicador. |
| Desempeño aceptable | 60% | Demuestra un desempeño competente, evidenciando el logro de los elementos básicos del indicador, pero con omisiones, dificultades y/o errores. |
| Desempeño incipiente | 30% | Presenta importantes omisiones, dificultades y/o errores en el desempeño, que no permite evidenciar los elementos básicos del logro del indicador, por lo que no puede ser considerado competente. |
| Desempeño no logrado | 0% | Presenta ausencia o incorrecto desempeño. |

| Indicador de logro | Indicador de evaluación | Categoría de Respuesta | | | | Ponderación del indicador de logro |
|---|---|---|---|---|--|------------------------------------|
| | | Muy buen desempeño 100% | Desempeño aceptable 60% | Desempeño incipiente 30% | Desempeño no logrado 0% | |
| IL1.1 Ejecuta los procedimientos fundamentales de carga de datos, prueba de funciones de activación y entrenamiento de la red neuronal aplicando los conceptos fundamentales de Deep Learning, tales como Red Neuronal Artificial, Perceptrón y Perceptrón multicapa. | 1.1.1 Realiza la carga de los datos de MNIST correctamente, para la implementación de una red neuronal, identificando la data de entrenamiento y de prueba. | Ejecuta la carga de los datos de MNIST correctamente, para la implementación de una red neuronal, identificando la data de entrenamiento y de prueba. | Ejecuta la carga de los datos de MNIST correctamente, para la implementación de una red neuronal, identificando la data de entrenamiento, pero no identifica la data de prueba. | Ejecuta con dificultad la carga de los datos de MNIST, para la implementación de una red neuronal, identificando en forma parcial la data de entrenamiento y la data de prueba. | No ejecuta la carga de los datos de MNIST correctamente, para la implementación de una red neuronal, identificando la data de entrenamiento y de prueba. | 5% |
| | 1.1.2 Prueba las distintas funciones de activación, variando la cantidad de capas ocultas entre 1 y 3, manteniendo la cantidad de neuronas. | Demuestra las tres distintas funciones de activación: ReLU, Tanh y Sigmoid, variando la cantidad de capas ocultas entre 1 y 3, manteniendo la cantidad de neuronas. | Demuestra solo dos de las tres distintas funciones de activación: ReLU, Tanh y Sigmoid, variando la cantidad de capas ocultas entre 1 y 3, manteniendo la cantidad de neuronas. | Demuestra con dificultad las tres distintas funciones de activación: ReLU, Tanh y Sigmoid, no variando la cantidad de capas ocultas entre 1 y 3, manteniendo la cantidad de neuronas. | No demuestra las tres distintas funciones de activación: ReLU, Tanh y Sigmoid, variando la cantidad de capas ocultas entre 1 y 3, manteniendo la cantidad de neuronas. | 7,5% |
| | 1.1.3 Entrena las diferentes combinaciones utilizando 20 épocas completando la tabla de resultados obtenidos. | Entrena en forma satisfactoria las diferentes combinaciones utilizando 20 épocas, completando la tabla de resultados obtenidos con claridad. | Entrena en forma parcial las diferentes combinaciones utilizando 20 épocas y completando la tabla con algunos resultados. | Entrena con dificultad las diferentes combinaciones utilizando 20 épocas no completando la tabla de resultados. | No entrena en forma satisfactoria las diferentes combinaciones utilizando 20 épocas no completando la tabla de resultados. | 7,5% |

| | | | | | | |
|---|---|---|--|--|--|------------|
| <p>IL1.2 Selecciona funciones de activación, error y salida, aplicada a redes neuronales artificiales simples, de acuerdo con el caso de estudio planteado utilizando técnicas fundamentales de DL.</p> | <p>1.2.1 Identifica las diferencias en rendimientos con la variación de capas, neuronas por capa y función de activación, completando la información obtenida en los campos dispuestos en la tabla correspondiente.</p> | <p>Establece en forma satisfactoria las diferencias en rendimientos con la variación de capas, neuronas por capa y función de activación, completando la información obtenida en los campos dispuestos.</p> | <p>Establece en forma parcial las diferencias en rendimientos con la variación de capas, neuronas por capa y función de activación, completando con dificultad la información obtenida en los campos dispuestos.</p> | <p>Establece con dificultad las diferencias en rendimientos con la variación de capas, neuronas por capa y función de activación, no completando la información obtenida en los campos dispuestos.</p> | <p>No establece las diferencias en rendimientos con la variación de capas, neuronas por capa y función de activación, no completando la información obtenida en los campos dispuestos.</p> | <p>15%</p> |
| | <p>1.2.2 Justifica correctamente por qué hay diferencias entre la aplicación de una función de activación versus otra.</p> | <p>Evidencia correctamente por qué hay diferencias entre la aplicación de una función de activación versus otra.</p> | <p>Evidencia en forma parcial por qué hay diferencias entre la aplicación de una función de activación versus otra.</p> | <p>Evidencia con dificultad por qué hay diferencias entre la aplicación de una función de activación versus otra.</p> | <p>No evidencia por qué hay diferencias entre la aplicación de una función de activación versus otra.</p> | <p>15%</p> |

| | | | | | | |
|---|--|---|---|--|--|----|
| IL1.3 Programa algoritmos en lenguaje Python, para optimizar un modelo de red neuronal artificial (MLP), utilizando Tensorflow mediante Keras en el contexto de un caso de estudio. | 1.3.1 Aplica Dropout al modelo del caso planteado haciendo carga de los datos de MNIST y entrenando la red. | Utiliza en forma correcta Dropout al modelo del caso planteado haciendo carga de los datos de MNIST y entrenando la red. | Utiliza en forma parcial Dropout al modelo del caso planteado haciendo carga de los datos de MNIST y entrenando con dificultad la red. | Utiliza con dificultad Dropout al modelo del caso planteado haciendo carga de los datos de MNIST, pero no entrenando la red. | No utiliza en forma correcta Dropout al modelo del caso planteado no haciendo carga de los datos de MNIST y no entrenando la red. | 6% |
| | 1.3.2 Mejora la generalización de la red neuronal, entre la primera y la mejor ejecución justificando y explicando los ajustes realizados. | Enriquece correctamente la generalización de la red neuronal, entre la primera y la mejor ejecución justificando y explicando los ajustes realizados. | Enriquece parcialmente la generalización de la red neuronal, entre la primera y la mejor ejecución justificando y explicando los ajustes realizados con dificultad. | Enriquece con dificultad la generalización de la red neuronal, entre la primera y la mejor ejecución no justificando ni explicando los ajustes realizados. | No la generalización de la red neuronal, entre la primera y la mejor ejecución no justificando ni explicando los ajustes realizados. | 6% |

| | | | | | |
|--|---|---|---|---|----|
| 1.3.3 Utiliza correctamente el método EARLY STOPPING y explican el impacto de incorporarlo en la arquitectura. | Emplea correctamente el método EARLY STOPPING y explican el impacto de incorporarlo en la arquitectura. | Emplea parcialmente el método EARLY STOPPING y explican medianamente el impacto de incorporarlo en la arquitectura. | Emplea con dificultad el método EARLY STOPPING y no explican el impacto de incorporarlo en la arquitectura. | No emplea con dificultad el método EARLY STOPPING y no explican el impacto de incorporarlo en la arquitectura. | 6% |
| 1.3.4 Fundamenta cuál es el rol de un optimizador dentro de un modelo de red neuronal según los recursos de aprendizaje asociados a la actividad y mejora el rendimiento de la red neuronal justificando y explicando los ajustes realizados con el uso de diferentes optimizadores. | Establece correctamente el rol de un optimizador dentro de un modelo de red neuronal según los recursos de aprendizaje asociados a la actividad y mejora el rendimiento de la red neuronal justificando y explicando los ajustes realizados con el uso de diferentes optimizadores. | Establece en forma parcial el rol de un optimizador dentro de un modelo de red neuronal según los recursos de aprendizaje asociados a la actividad y no mejora el rendimiento de la red neuronal justificando y explicando los ajustes realizados con el uso de diferentes optimizadores. | Establece con dificultad el rol de un optimizador dentro de un modelo de red neuronal según los recursos de aprendizaje asociados a la actividad y no mejora el rendimiento de la red neuronal justificando y explicando los ajustes realizados con el uso de diferentes optimizadores. | No establece correctamente el rol de un optimizador dentro de un modelo de red neuronal según los recursos de aprendizaje asociados a la actividad y no mejora el rendimiento de la red neuronal justificando y explicando los ajustes realizados con el uso de diferentes optimizadores. | 6% |

| | | | | | | |
|--|--|---|--|--|--|-----|
| | 1.3.5 Fundamenta la diferencia entre un optimizador con tasa de aprendizaje (learning rate) constante versus un optimizador con learning rate adaptativo según los contenidos abordados. | Establece correctamente la diferencia entre un optimizador con tasa de aprendizaje (learning rate) constante versus un optimizador con learning rate adaptativo según los contenidos abordados. | Establece en forma parcial la diferencia entre un optimizador con tasa de aprendizaje (learning rate) constante versus un optimizador con learning rate adaptativo según los contenidos abordados. | Establece con dificultad la diferencia entre un optimizador con tasa de aprendizaje (learning rate) constante versus un optimizador con learning rate adaptativo según los contenidos abordados. | No establece la diferencia entre un optimizador con tasa de aprendizaje (learning rate) constante versus un optimizador con learning rate adaptativo según los contenidos abordados. | 6% |
| IL1.4 Evalúa el desempeño de los modelos utilizando métricas de acuerdo con el caso de estudio presentado. | 1.4.1 Documenta los resultados obtenidos en las tablas dispuestas, con el objetivo de facilitar y visibilizar los diferentes índices. | Registra correctamente los resultados obtenidos en las tablas dispuestas, con el objetivo de facilitar y visibilizar los diferentes índices. | Registra parcialmente los resultados obtenidos en las tablas dispuestas, con el objetivo de facilitar y visibilizar los diferentes índices. | Registra con dificultad los resultados obtenidos en las tablas dispuestas, con el objetivo de facilitar y visibilizar los diferentes índices. | No registra con los resultados obtenidos en las tablas dispuestas, con el objetivo de facilitar y visibilizar los diferentes índices. | 10% |

| | | | | | |
|---|--|---|---|--|-----|
| 1.4.2 Genera gráficos que ayudan a comparar las métricas obtenidas usando librerías de visualización como matplotlib o seaborn. | Produce correctamente gráficos que ayudan a comparar las métricas obtenidas usando librerías de visualización como matplotlib o seaborn. | Produce en forma parcial gráficos que ayudan a comparar las métricas obtenidas usando librerías de visualización como matplotlib o seaborn. | Produce con dificultad gráficos que ayudan a comparar las métricas obtenidas usando librerías de visualización como matplotlib o seaborn. | No produce gráficos que ayudan a comparar las métricas obtenidas usando librerías de visualización como matplotlib o seaborn. | 10% |
| Dimensión Individual (FORMATIVA) | | | | | |
| Responde las preguntas realizadas por el/la docente respecto a los aspectos teóricos que fundamentan el diseño y programación del proyecto de DL. | Responde de manera correcta y completa las preguntas realizadas por el/la docente. | Responde las preguntas realizadas por el/la docente de manera parcial. | Responde las preguntas, pero de manera incompleta o imprecisa. | No responde preguntas del docente. | - |
| Utiliza lenguaje técnico para presentar el proyecto de DL. | Utiliza lenguaje técnico y formal para presentar proyecto | Utiliza lenguaje técnico y formal, pero incurre en algunas ocasiones en el uso de modismos e informalidades. | Incurre en el uso de lenguaje coloquial durante la exposición en reiteradas ocasiones. | El lenguaje utilizado en la presentación no responde a los aspectos técnicos disciplinares, evidenciando poco dominio del proyecto desarrollado. | - |

