

Tarea 2 – Aprendizaje Automático 2024-1

Redes neuronales con QuickDraw Dataset.

Nícolas Farfán Cheneaux

Este estudio examina la efectividad de distintas configuraciones de redes neuronales multicapa (MLP) en la clasificación de imágenes del conjunto de datos QuickDraw, utilizando tres modelos con variaciones en la profundidad de las capas, las funciones de activación y las funciones de pérdida. El **Modelo 1** combina capas con activaciones ReLU y ELU, usando entropía cruzada categórica como función de pérdida. El **Modelo 2** integra tres capas con activación ReLU, también empleando entropía cruzada categórica, y el **Modelo 3** alterna entre activaciones ReLU y ELU en cuatro capas, aplicando la divergencia de Kullback-Leibler. Cada modelo fue entrenado y evaluado repetidamente para asegurar la consistencia de los resultados frente a la variabilidad en la inicialización de pesos. Se utilizó la precisión por clase y la precisión total para evaluar los modelos, representadas a través de matrices de confusión agregadas. Los hallazgos indican que la adecuada selección de funciones de activación y pérdida, junto con la estructura del modelo, juegan un papel crucial en la precisión de la clasificación, revelando que **aumentar la complejidad del modelo no necesariamente conduce a un mejor rendimiento**.

1. Introducción, descripción del problema y contexto

En el campo del aprendizaje automático, la clasificación de imágenes constituye uno de los problemas fundamentales, especialmente con el creciente volumen y variedad de datos visuales disponibles. Este desafío se vuelve aún más prominente con conjuntos de datos como QuickDraw, una colección extensa de dibujos hechos a mano por usuarios de todo el mundo. Estos dibujos, que capturan un boceto de objetos cotidianos con trazos simplificados, presentan una oportunidad única para explorar cómo las redes neuronales pueden interpretar y clasificar representaciones gráficas abstractas.

Las redes neuronales multicapa (MLP), específicamente, son una clase de modelos que han demostrado ser efectivas en tareas de clasificación debido a su capacidad para aprender representaciones jerárquicas de datos. Sin embargo, el diseño de estas redes—incluyendo la selección de la arquitectura, las funciones de activación y las funciones de pérdida—puede variar significativamente, influenciando directamente su rendimiento en tareas específicas.

El propósito de este estudio es investigar cómo diferentes configuraciones de MLP afectan la precisión de clasificación en el conjunto de datos QuickDraw. A través de experimentos con tres modelos distintos, este trabajo busca determinar qué combinaciones de número de capas, tipos de activación y funciones de pérdida resultan más efectivas para esta tarea única de clasificación. Los modelos seleccionados varían en complejidad y tipo de activación, desde configuraciones básicas con ReLU hasta arquitecturas más complejas que incorporan ELU y funciones de pérdida avanzadas como la divergencia de Kullback-Leibler.

Este análisis no solo contribuirá a un mejor entendimiento de cómo las redes neuronales pueden ser adaptadas a conjuntos de datos de dibujos, sino que también proporcionará insights valiosos sobre el impacto de la arquitectura de red en la clasificación de imágenes, un área de interés permanente en y el aprendizaje profundo.

2. Desarrollo

2.1 Diseño de Hiperparámetros

Modelo 1:

- **Capas y Neuronas:** Se utilizó una configuración con 128 y 64 neuronas en las capas ocultas. Esta estructura proporciona un equilibrio entre simplicidad y suficiente capacidad para capturar características esenciales sin caer en el sobreajuste.
- **Funciones de Activación:** ReLU se eligió para la primera capa oculta debido a su eficiencia y efectividad en evitar el problema del gradiente desvanecido en las primeras etapas del aprendizaje. ELU se utilizó en la segunda capa para introducir capacidades de normalización y contrarrestar algunos de los problemas potenciales de ReLU, como las neuronas "muertas".
- **Función de Pérdida:** La entropía cruzada categórica es estándar para problemas de clasificación multiclase, ya que mide la discrepancia entre las etiquetas verdaderas y las probabilidades predichas, proporcionando un entrenamiento eficaz y directo.

Modelo 2:

- **Capas y Neuronas:** Se aumentó la profundidad y el número de neuronas (256, 128, 64) para explorar

si una red más compleja podría capturar mejor la variabilidad y la riqueza del conjunto de datos QuickDraw, potencialmente mejorando la precisión.

- **Funciones de Activación:** Todas las capas ocultas usan ReLU, aprovechando su capacidad para acelerar la convergencia del entrenamiento y manejar efectivamente los gradientes en redes más profundas.
- **Función de Pérdida:** Se mantuvo la entropía cruzada categórica, favoreciendo la coherencia en la evaluación entre modelos con diferente arquitectura, pero bajo el mismo tipo de tarea de clasificación.

Modelo 3:

- **Capas y Neuronas:** Este modelo alterna entre 128 neuronas con activaciones ReLU y ELU en cuatro capas. Esta configuración busca un equilibrio avanzado entre evitar neuronas muertas y mejorar la capacidad del modelo para ajustarse a los datos sin ajustarse demasiado al ruido.
- **Funciones de Activación:** La combinación de ReLU y ELU tiene como objetivo beneficiar de ambos mundos: ReLU para una propagación eficiente del gradiente y ELU para incluir cierta respuesta a variaciones negativas en las entradas.
- **Función de Pérdida:** Se seleccionó la divergencia de Kullback-Leibler para este modelo más complejo, buscando explorar cómo esta métrica, que mide cómo una probabilidad modelo se desvía de la verdadera distribución de probabilidad, podría influir en la capacidad del modelo para generalizar más allá del simple ajuste de datos.

2.2 Implementación de la Función de Entrenamiento

La función `train_model` fue diseñada para entrenar cada uno de los modelos especificados en múltiples iteraciones para asegurar la robustez y reproducibilidad de los resultados. Durante cada iteración, se realizan las siguientes acciones:

- **Inicialización y Entrenamiento:** Cada modelo se compila y entrena utilizando imágenes y etiquetas del conjunto de entrenamiento. El entrenamiento se lleva a cabo sin salida verbal (`verbose=0`) para mantener la limpieza del proceso.

- **Evaluación:** Posterior al entrenamiento, cada modelo se evalúa utilizando un conjunto de pruebas separado, donde se calcula la pérdida y la precisión.
- **Predicción y Matriz de Confusión:** Se realizan predicciones sobre el conjunto de pruebas, y con estos resultados, se genera una matriz de confusión para cada iteración. Las predicciones también se acumulan para análisis agregados.
- **Visualización:** Se generan y visualizan gráficos de barras para la precisión por clase y matrices de confusión para proporcionar interpretaciones visuales de la precisión y el comportamiento del modelo.

Consideraciones Adicionales

- **Reproducibilidad:** Cada modelo utiliza una inicialización aleatoria controlada para garantizar que los resultados sean reproducibles y comparables entre diferentes ejecuciones.
- **Optimización:** Se utiliza el optimizador Adam por su eficacia demostrada en numerosos problemas de aprendizaje profundo, ajustando automáticamente la tasa de aprendizaje durante el entrenamiento.

3. Resultados / Experimentación

3.1 Experimentación con funciones de activación

ReLU (Rectified Linear Unit)

- Para los Modelos 1, 2 y 3, ReLU se utilizó como una función de activación primaria en varias capas ocultas para aprovechar su capacidad de mantener gradientes saludables a lo largo de muchas épocas de entrenamiento, facilitando la convergencia rápida y evitando las complicaciones del entrenamiento en redes más profundas.

ELU (Exponential Linear Unit)

- En el Modelo 1 y el Modelo 3, ELU se incorporó para explorar estos beneficios, particularmente para manejar mejor las activaciones negativas y proporcionar un comportamiento más robusto en capas donde la saturación de neuronas podría ser un problema. Este enfoque fue particularmente pertinente en configuraciones donde la complejidad aumentaba, proporcionando un

balance entre la rápida activación de ReLU y la respuesta más controlada de ELU.

3.2 Experimentación con el número de capas

Con respecto al modelo 1 la reducción gradual del número de neuronas ayuda a prevenir el sobreajuste, especialmente cuando el conjunto de datos no es extremadamente grande o complejo. Esta configuración proporciona suficiente capacidad para aprender las distinciones básicas entre clases sin incurrir en una complejidad computacional excesiva. Por otro lado, en el modelo 2 el incremento en el número de neuronas en la primera capa permite al modelo explorar un espacio de características más amplio, mientras que las reducciones sucesivas ayudan a concentrar estas características en predicciones más precisas. Esta estructura está pensada para mejorar la capacidad del modelo de generalizar bien a nuevas imágenes, manteniendo un control sobre el riesgo de sobreajuste y finalmente en el modelo 3 mantener un número constante de neuronas a través de múltiples capas puede ayudar a estabilizar el aprendizaje en redes profundas, evitando los efectos negativos de los cambios abruptos en la dimensionalidad. La repetición de la estructura intenta profundizar la capacidad de abstracción del modelo sin aumentar demasiado la cantidad de parámetros, equilibrando así el aprendizaje y la generalización.

3.3 Experimentación con hiperparámetro loss function

Modelos 1 y 2: Categorical Crossentropy

- **Categorical Crossentropy:** Es la elección estándar para problemas de clasificación multiclase. Esta función mide la diferencia entre las distribuciones de probabilidad real y prevista, proporcionando una evaluación clara de cuán bien el modelo predice la clase correcta en cada caso. Al minimizar esta función de pérdida, el modelo ajusta sus parámetros para producir la mejor estimación de la probabilidad de cada clase.

Modelo 3: KLDivergence

- **KLDivergence (Divergencia de Kullback-Leibler):** Esta función se utiliza para medir cómo una distribución de probabilidad se diferencia de otra referencia. Es útil en escenarios donde las distribuciones de las clases no son uniformes o cuando se desea un modelo que pueda aproximarse de manera más precisa a la distribución real de los datos en el conjunto de entrenamiento.

3.4 Resultados obtenidos

Se hicieron múltiples pruebas con varias combinaciones de hiperparámetros como el número de capas y distintas funciones de activación como **sigmoid y tan**, funciones de pérdida mencionadas y épocas entre 10 y 40, y dado lo fundamentado anteriormente se decidió por mantener los modelos planteados, obteniendo los resultados en cuanto a:

Robustez: Se ha observado que se tiene un mejor resultado en Quickdraw-10 dado que en ocasiones en modelo 3 tiene un buen desempeño llegando al 50%, pero en ocasiones se sobreajusta. Por lo que se puede concluir que los modelos 1 y 2 son robustos al tener resultados similares para ambos datasets.

Tiempo: En promedio con las épocas determinadas (20) se ha obtenido un tiempo medio de ejecución estimado de 6 minutos por MLP. Lo cual es aceptable al estar usando un equipo de condiciones estándar con una gráfica NVIDIA gtx 1650, 16gb de RAM e Intel Core i5 9X.

Calidad: En cuanto a la calidad, la clave está en las funciones de activación dado que es necesario comprender cuando es ideal utilizar cada una, por ejemplo, la función ELU y ReLU en este caso tienen un buen desempeño, pero funciones como sigmoid y tan no lo tienen debido al conjunto de datos.

Resultados experimentales Dataset Quickdraw-10

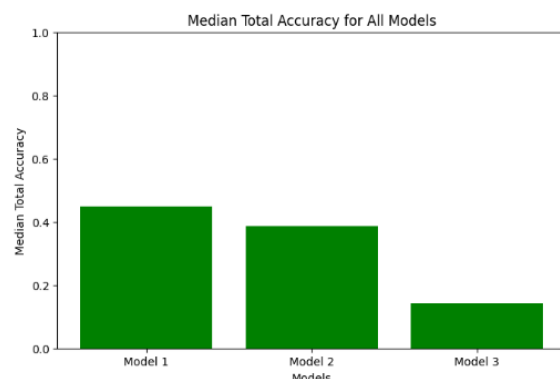


Ilustración 1: Precisión mediana total.

Se puede apreciar que se obtiene una precisión usando la mediana donde el modelo 1 y 2 cercano al 50%. Sin embargo, el modelo nº 3 no tuvo un buen desempeño, esto se debe a que está sobre ajustado como se puede apreciar en la matriz de confusión y en el gráfico de barras (ver código).

Y se tiene la precisión de la mediana para cada clase de todas las iteraciones

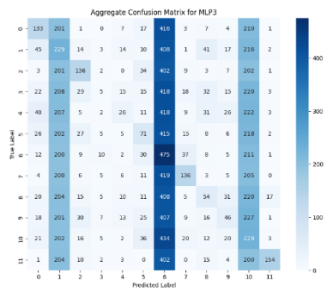


Ilustración 2: Matriz de confusión MLP 3 QuickDraw-10

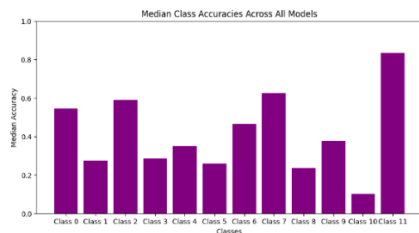


Ilustración 3: Precisión mediana por clase. QuickDraw-10

Resultados experimentales Dataset Quickdraw Animals

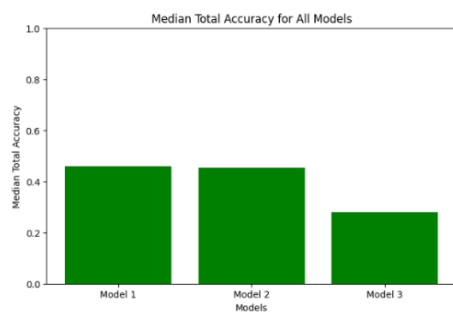


Ilustración 4: Precisión mediana total. QuickDraw Animals

Se puede apreciar un resultado muy parecido al dataset de Quick-Draw 10. Esto podría deberse a que las imágenes no son tan distintas y la red se comporta de manera similar, esto queda más claro con el siguiente gráfico.

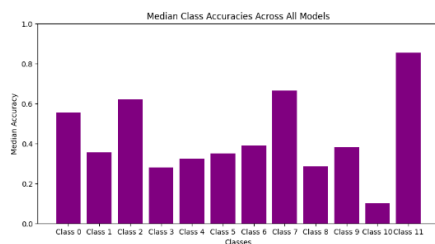


Ilustración 5: Precisión mediana por clase. QuickDraw Animals

Donde otra vez se tiene una mediana de precisión parecida.

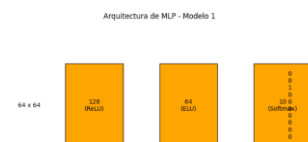


Ilustración 6: Diagrama de bloques MLP 1

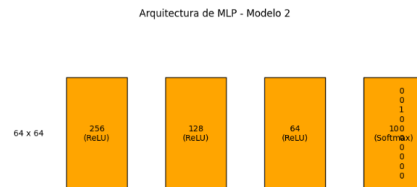


Ilustración 7: Diagrama de bloques MLP 2

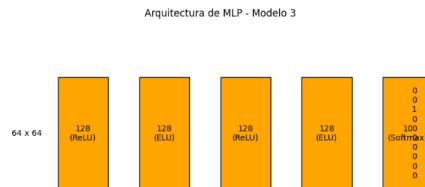


Ilustración 8: Diagrama de bloques MLP 3

En cuanto a ventajas y desventajas los modelos 1 y 2 son robustos y tienen un desempeño uniforme para ambos dataset, la desventaja es el modelo 3 que dadas las múltiples pruebas con 4 capas se está sobre ajustando por lo que lo ideal son menor a 3 capas con las mismas funciones de activación funcionales para este caso (elu y relu).

4. Conclusiones

Los experimentos realizados con distintos modelos de redes neuronales multicapa (MLP) para la clasificación de imágenes del conjunto de datos QuickDraw demuestran que la estructura del modelo y la elección de funciones de activación y pérdida son determinantes para su rendimiento. El Modelo 1, con dos capas ocultas y una combinación de activaciones **ReLU** y **ELU**, logró una precisión cercana al 50%, mostrando un buen equilibrio entre simplicidad y capacidad de generalización. El Modelo 2, más profundo con tres capas ocultas y activaciones **ReLU**, también alcanzó una precisión similar, lo que sugiere que aumentar la complejidad puede mejorar la captura de características sin sobre ajustarse. Sin embargo, el Modelo 3, con cuatro capas y una alternancia de activaciones ReLU y ELU junto con la función de pérdida de divergencia de Kullback-Leibler, no tuvo un buen desempeño, indicando que la mayor complejidad llevó a un sobreajuste. En resumen, una mayor complejidad del modelo no siempre garantiza un mejor rendimiento, y la selección cuidadosa de activaciones y funciones de pérdida es crucial para obtener una buena precisión dada la experimentación con funciones de activación más simples como sigmoid.