Apuntes de Clase 1 S14: Cuantificación y Aprendizaje No Supervisado

John Sánchez C.

Tecnológico de Costa Rica
Correo Electrónico: jostsace@estudiantec.cr

***Resumen*—Este documento resume los conceptos clave abordados en la semana 14 del curso, enfocándose en las técnicas de cuantificación y los fundamentos del aprendizaje no supervisado. Se exploran las ventajas de la cuantificación, como la reducción del tamaño del modelo y la mejora en la eficiencia computacional y energética, detallando los enfoques asimétrico y simétrico, así como las estrategias de cuantificación dinámica, post-entrenamiento (PTQ) y consciente del entrenamiento (QAT). Además, se introduce el aprendizaje no supervisado, cubriendo la reducción de dimensionalidad mediante autoencoders y las diversas técnicas de clustering, incluyendo sus principios, métricas de similitud y tipos. El objetivo es proporcionar una visión clara de cómo estas técnicas optimizan los modelos de aprendizaje automático y permiten el descubrimiento de patrones en datos no etiquetados.**

I. Anuncios

1. **Proyecto 3 y Tarea 3** Se informó sobre la posible publicación del Proyecto 3 y Tarea 3 para el jueves de esta semana, aunque finalmente no se realizaron.
2. **Revisión del Proyecto 2** La revisión del Proyecto 2 está programada para la semana 15.

II. Introducción

La cuantificación (quantization) es una técnica fundamental en la optimización de modelos de aprendizaje automático, cuyo objetivo principal es reducir la precisión numérica de los parámetros del modelo, típicamente de punto flotante a representaciones de enteros de menor bit. Este proceso conlleva múltiples beneficios, incluyendo una significativa disminución en el tamaño de los modelos, lo que facilita su almacenamiento y despliegue en dispositivos con recursos limitados. Además, la cuantificación acelera el tiempo de inferencia, ya que las operaciones con números enteros son inherentemente más rápidas y energéticamente eficientes que las operaciones con punto flotante, especialmente en hardware optimizado para enteros. Esta introducción sentará las bases para comprender cómo la cuantificación impacta positivamente la eficiencia de los modelos y cómo se integra con el aprendizaje no supervisado para mejorar el procesamiento de datos.

III. Repaso - Cuantificación (Parte 1)

La cuantificación es el proceso de reducir el número de bits necesarios para representar cada parámetro (pesos y activaciones) de un modelo de aprendizaje automático, convirtiendo valores de punto flotante a representaciones enteras de menor precisión. Es crucial entender que esto no es simplemente un redondeo; implica un mapeo matemático que busca preservar la mayor cantidad de información posible.

III-A. Ventajas de la Cuantificación

La aplicación de técnicas de cuantificación ofrece beneficios sustanciales en el despliegue de modelos:

- **Menor consumo de memoria:** Al reducir el tamaño de los parámetros, el modelo ocupa menos espacio en la memoria, facilitando su ejecución en dispositivos con recursos limitados.
- **Menor tiempo de inferencia:** Las operaciones con enteros son computacionalmente menos costosas que las de punto flotante, lo que se traduce en una ejecución más rápida del modelo.
- **Menor consumo de energía:** La eficiencia computacional lograda por la cuantificación contribuye a una reducción en el consumo de energía, aspecto crítico en aplicaciones móviles y embebidas.

III-B. Cuantificación Asimétrica

La cuantificación asimétrica mapea un rango de valores de punto flotante $[\beta, \alpha]$ a un rango de enteros no negativos $[0, 2^n - 1]$, donde n es el número de bits utilizados para la representación.

- β: Corresponde al valor mínimo (más bajo) en el rango de los valores a cuantificar.
- α: Corresponde al valor máximo en el rango de los valores a cuantificar.

Por ejemplo, utilizando 8 bits, se puede representar un total de $2^8 = 256$ valores, mapeados al rango $[0, 255]$. A continuación se presentan las fórmulas para la cuantificación y descuantificación asimétrica:

- **Fórmula de cuantificación:** $x_q = \text{clamp}\left(\left\lfloor \frac{x_f}{s} \right\rfloor + z;\ 0;\ 2^n - 1\right)$
  - x_f: Valor flotante a cuantificar.
- **Parámetro de escalado s:** Este parámetro define la escala a la que se mapean los valores flotantes.
  - $s = \frac{\alpha - \beta}{2^n - 1}$
  - $2^n - 1$: Representa el rango máximo del espacio de salida cuantificado.
- **Parámetro de punto cero z:** Este es el desplazamiento o punto cero, que asegura que el valor β en punto flotante se mapee a 0 en el espacio entero.

- $z = \left\lfloor -1 \cdot \frac{\beta}{s} \right\rceil$

- n: El número de bits utilizados para la cuantificación.

Para realizar la **des-cuantificación asimétrica**, que convierte un valor entero cuantificado de vuelta a un valor flotante aproximado, se utiliza la siguiente fórmula. Es importante destacar que este proceso introduce un grado de error inherente debido a la pérdida de precisión durante la cuantificación.

- **Fórmula de des-cuantificación:** $x_f = s(x_q - z)$

III-C. Cuantificación Simétrica

A diferencia de la cuantificación asimétrica, la cuantificación simétrica mapea valores entre un rango simétrico de $[-\alpha, \alpha]$ a un rango de enteros que también es simétrico, típicamente $[-(2^{n-1}-1), (2^{n-1}-1)]$. Aquí, α representa el valor absoluto máximo en el rango de los datos. Por ejemplo, con 8 bits, el rango entero puede ser $[-127, 127]$. A continuación se presentan los pasos para realizar una cuantificación simétrica:

- **Fórmula de cuantificación:** $x_q = \text{clamp}\left(\left\lfloor \frac{x_f}{s} \right\rceil ; -(2^{n-1}-1); 2^{n-1}-1\right)$
- **Parámetro de escalado s:**
  - $s = \frac{\text{abs}(\alpha)}{2^{n-1}-1}$
- n: El número de bits utilizados.

Para realizar la **des-cuantificación simétrica**, el proceso es más sencillo:

- **Fórmula de des-cuantificación:** $x_f = sx_q$

III-D. Cuantificación Dinámica (Dynamic Quantization)

La cuantificación dinámica es una estrategia que aplica cuantificación de forma estática para los pesos del modelo, es decir, los pesos se cuantifican antes de la inferencia y permanecen fijos. Sin embargo, para las activaciones, la cuantificación se realiza "sobre la marcha"(just-in-time) durante la inferencia del modelo. Esto permite que el rango de cuantificación para las activaciones se determine dinámicamente en función de los valores reales que toman durante la ejecución, lo que puede mejorar la precisión en comparación con la cuantificación estática de activaciones.

III-E. Calibración (Calibration)

La calibración es una etapa crítica en la cuantificación post-entrenamiento. Se lleva a cabo realizando inferencias con un conjunto de datos de çalibración"(que deben ser representativos de los datos de entrada esperados, pero nunca antes vistos por el modelo durante el entrenamiento). Durante esta inferencia, se observan y recopilan estadísticas (como los valores mínimos y máximos o histogramas) de las activaciones en cada capa. Esta información se utiliza luego para determinar los parámetros s (escalado) y z (punto cero) óptimos que permitirán cuantificar el modelo de manera efectiva. Se realiza específicamente en la etapa de cuantificación post-entrenamiento (PTQ).

III-F. Selección del Rango

La elección del rango de cuantificación (los valores de α y β) para ambos tipos de cuantificación es crucial y puede introducir un grado significativo de error, especialmente en presencia de valores atípicos (outliers). Los outliers pueden distorsionar el rango, forzando a que los valores más comunes se mapeen a un subconjunto más pequeño del rango cuantificado, lo que reduce la precisión. Una solución común para mitigar este problema es utilizar una estrategia basada en el percentil de la distribución del vector de valores. Al ignorar un pequeño porcentaje de los valores más extremos (por ejemplo, el $0.1\,\%$ inferior y superior), se reduce la sensibilidad a los outliers y se obtiene un rango de cuantificación más robusto y representativo.

III-G. Granularidad de Cuantificación (en Convolución)

En el contexto de las redes neuronales convolucionales, se ha observado que aplicar la cuantificación de forma más granular mejora la precisión. En lugar de aplicar un único par de valores β y α (o α absoluto para simétrica) para toda una capa, es más beneficioso aplicar la cuantificación por filtro individualmente. Esto se debe a que los diferentes filtros dentro de una capa convolucional pueden tener distribuciones de activaciones y pesos muy distintas, y por lo tanto, requieren sus propios parámetros de escalado (s) y punto cero (z) óptimos. Al personalizar estos parámetros para cada filtro, se logra una cuantificación más precisa y se minimiza la pérdida de información.

IV. Cuantificación (Parte 2)

IV-A. Cuantificación Post-Entrenamiento (Post-Training Quantization - PTQ)

El PTQ es una técnica de cuantificación que se aplica a un modelo que ya ha sido completamente entrenado con precisión de punto flotante. Los valores de los pesos del modelo se convierten a una representación de menor precisión (generalmente enteros) después de que el entrenamiento ha concluido. Para determinar los parámetros de cuantificación (s y z), se utiliza un pequeño conjunto de datos de calibración (nunca antes visto por el modelo durante el entrenamiento) para recopilar estadísticas de las activaciones de cada capa, como sus rangos o distribuciones. Basándose en esta información, se calculan los parámetros s y z que permiten cuantificar el modelo. Este método es atractivo por su simplicidad y porque no requiere reentrenar el modelo.

IV-B. Cuantificación Consciente del Entrenamiento (Quantization Aware Training - QAT)

QAT es un enfoque más avanzado donde la cuantificación se simula durante el proceso de entrenamiento del modelo. En lugar de cuantificar el modelo después de su entrenamiento, se insertan "módulos de cuantificación falsos"(o "fake quantization"nodes) en la red. Estos módulos simulan el efecto de la cuantificación de enteros en los pasos de avance y retroceso del entrenamiento. Esto significa que los gradientes se calculan a través de operaciones que emulan la cuantificación,

permitiendo que la función de pérdida del modelo ”sientaz se adapte a los errores introducidos por la cuantificación. Como resultado, los pesos del modelo se actualizan constantemente mientras ”sufren.[el] efecto de la cuantificación, generando un modelo intrínsecamente más robusto y tolerante a la pérdida de precisión.

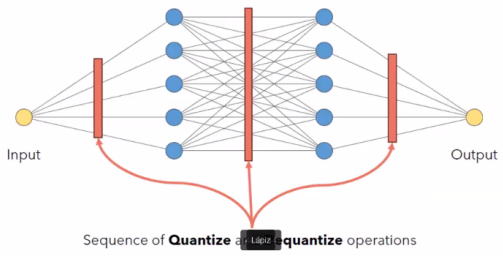

Figura 1. Representación esquemática de una red neuronal con *observers*. Las líneas rojas indican los puntos donde se monitorean las activaciones para recopilar estadísticas en el proceso de cuantificación.

En la Figura 1 se ilustra una red neuronal donde los .[o]bservers”, representados por las líneas rojas, son los puntos dentro de la arquitectura donde se recopilan las estadísticas de las activaciones para la calibración o para la simulación de cuantificación.

V. Aprendizaje No Supervisado

El aprendizaje no supervisado es una rama del aprendizaje automático que se ocupa de encontrar patrones y estructuras en conjuntos de datos que no están etiquetados. A diferencia del aprendizaje supervisado, donde se utiliza un conjunto de datos con etiquetas conocidas para entrenar el modelo, el aprendizaje no supervisado busca descubrir relaciones intrínsecas, agrupaciones o representaciones de menor dimensionalidad sin ninguna guía externa.

V-A. Reducción de la Dimensionalidad

La reducción de dimensionalidad es un conjunto de técnicas utilizadas para reducir el número de características (dimensiones) en un conjunto de datos, manteniendo la mayor cantidad de información posible. Los autoencoders son un ejemplo prominente de modelos que realizan esta tarea.

- **Autoencoders:** Son redes neuronales que aprenden una representación eficiente (un .[es]pacio latente.[o] .[em]bedding”) de los datos de entrada. Funcionan codificando la entrada en una representación de menor dimensión y luego decodificando esa representación para reconstruir la entrada original. Este proceso obliga al autoencoder a aprender las características más importantes de los datos.
- **Aplicaciones:** La representación en espacios latentes (embeddings) es útil para:
  - **Visualización:** Permite proyectar datos de alta dimensión en 2D o 3D para su inspección.
  - **Detección de anomalías:** Los puntos que no se reconstruyen bien pueden ser considerados anomalías, ya que no se ajustan a los patrones aprendidos en el espacio latente.
  - **Compresión de datos:** Los embeddings pueden ser una representación compacta de los datos originales.

V-B. Clustering

El clustering (agrupamiento) es un procedimiento para identificar grupos naturales (clusters) en un conjunto de datos, basándose en la similitud de las características de los puntos de datos. Su objetivo es dividir el espacio del conjunto de datos en N grupos, de manera que los puntos dentro de cada grupo sean lo más homogéneos posible y los puntos de diferentes grupos sean lo más heterogéneos posible.

Existe una gran variedad de algoritmos de clustering, y la elección del más adecuado depende de la forma de los datos y de los patrones que se desean identificar. A continuación, se presenta una imagen que ilustra diversos algoritmos de agrupamiento y las formas de datos que pueden manejar eficazmente:

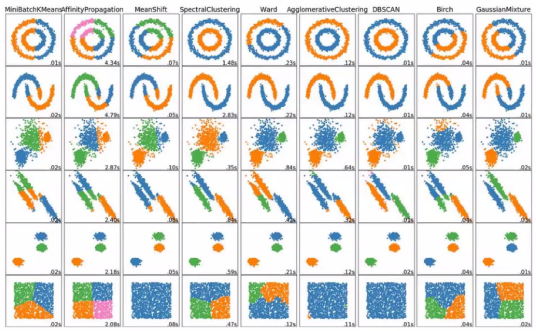

Figura 2. Ejemplos de diferentes algoritmos de clustering y las formas de datos que pueden agrupar eficientemente.

V-C. Clustering - Principios y Técnicas

Los algoritmos de clustering se basan en varios principios y técnicas fundamentales:

- **Similaridad o Distancia:** Para medir cuán çercanos.[o] ”similares”son los objetos en un conjunto de datos, típicamente se utiliza una métrica de distancia. Cuanto menor sea la distancia entre dos puntos, mayor será su similitud. Es importante notar que las métricas de distancia tradicionales no son adecuadas para datos categóricos. Algunas distancias comúnmente utilizadas incluyen: La Figura 3 muestra algunas de las métricas de distancia más empleadas en clustering.
- **Centralidad del Clúster:** Muchos algoritmos de clustering se basan en puntos centrales, conocidos como centroides (para algoritmos basados en la media) o medoides (para algoritmos basados en puntos reales del dataset), que representan el promedio o la ubicación central de todos los puntos dentro de un clúster. El centroide actúa como el ”líder.[o] la representación característica del grupo.
- **Minimización de la Función Objetivo:** La mayoría de los algoritmos de clustering buscan optimizar (minimizar

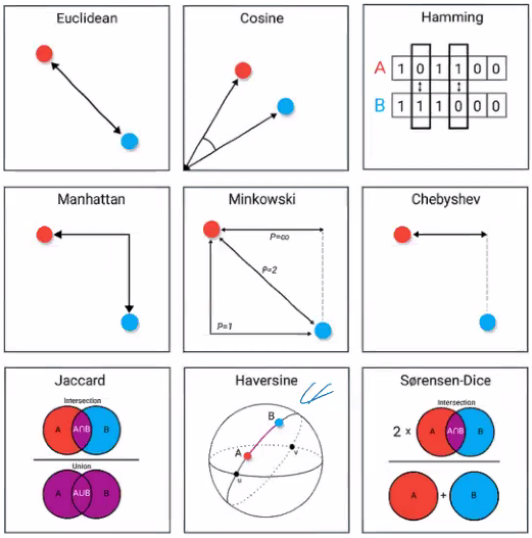

Figura 3. Ejemplos de métricas de distancia comunes utilizadas en algoritmos de clustering.

o maximizar) una función objetivo para determinar la mejor disposición de los clústeres. Por ejemplo:

- **K-means:** Minimiza la suma de los cuadrados de las distancias entre cada punto y el centroide de su clúster asignado (inercia).
- **Algoritmos basados en densidad (ej. DBSCAN):** Buscan maximizar la densidad de los puntos dentro de los clústeres.

V-D. Clustering - Tipos

Los algoritmos de clustering se pueden clasificar en varios tipos principales:

- **Jerárquico:** Este tipo de clustering construye una jerarquía de clústeres. Puede ser aglomerativo (cada muestra comienza en su propio clúster y los pares más similares se fusionan sucesivamente) o divisivo (todas las muestras comienzan en un solo clúster grande y se dividen en sub-clústeres). El proceso se itera hasta que se cumple una condición de parada, o se tiene un solo clúster (aglomerativo) o cada punto es un clúster individual (divisivo).
- **Determinístico (Hard Clustering):** En este tipo, cada muestra de datos se asigna de forma exclusiva a un único clúster. La muestra "pertenece.º "no pertenece.ª un clúster dado. Ejemplos incluyen K-means.
- **Probabilístico (Soft/Fuzzy Clustering):** A diferencia del determinístico, en el clustering probabilístico, cada muestra se asigna a uno o más clústeres con una cierta probabilidad o grado de pertenencia. Un ejemplo notable es Gaussian Mixture Models (GMM), donde cada punto tiene una probabilidad de pertenecer a cada distribución gaussiana que compone los clústeres.

VI. Conclusiones

La semana de clases abordó dos áreas fundamentales en el campo del aprendizaje automático: la cuantificación y el aprendizaje no supervisado. Se concluye que la cuantificación es una estrategia esencial para optimizar los modelos, permitiendo su despliegue eficiente en entornos con recursos limitados. La elección entre cuantificación asimétrica y simétrica, así como la aplicación de técnicas como PTQ o QAT, depende de los requisitos de precisión y el contexto de entrenamiento. Aunque introduce un grado de error, sus beneficios en tamaño y velocidad de inferencia son invaluables. Por otro lado, el aprendizaje no supervisado, con la reducción de dimensionalidad y el clustering, se presenta como una herramienta poderosa para descubrir estructuras ocultas y patrones en datos no etiquetados. La comprensión de sus principios y la correcta selección de algoritmos de clustering son cruciales para el análisis exploratorio de datos, la detección de anomalías y la segmentación, abriendo caminos para el conocimiento en ausencia de supervisión directa. La combinación de estas técnicas subraya la flexibilidad y la versatilidad de los modelos de aprendizaje automático en diversos escenarios.