Universidad del Valle de Guatemala
Facultad de ingeniería

Security Data Science
Proyecto 2

Entrenamiento Incremental en Modelos de Deep Learning y Machine Learning

Andrés de la Roca

Guatemala, 2024

Investigación Teórica

Típicamente el entrenamiento de los modelos de machine learning toma un enfoque de entrenamiento “total”, es decir que el modelo desarrollado se entrena sobre todo un dataset en una sola sesión. Este acercamiento es útil en ocasiones donde el conjunto de datos es pequeño y puede ser procesado de manera rápida, sin embargo, esta opción no es óptima para conjuntos de datos donde se tiene una cantidad grande datos como para ser procesados en una sola sesión o cuando los datos son cambiantes, como por ejemplo con datos de redes sociales o datos del mercado de la bolsa.
El entrenamiento incremental busca solventar este problema, creando oportunidades para que el modelo pueda aprender y mejorar progresivamente, esto se hace mediante la entrada de nuevos datos, es decir que para datos que se van actualizando constantemente se va a poder observar una adaptación correcta del modelo a estos nuevos datos, (Ade, Deshmukh., 2013) de cierta manera este enfoque hacia el aprendizaje es muy similar al de un humano, ya que el modelo va aprendiendo patrones y conectando conceptos previamente aprendidos incrementalmente para refinar las predicciones que está haciendo con los datos que se le dan.
Entre los beneficios principales que tiene el entrenamiento incremental están los siguientes:

- **Uso eficiente de recursos:**
  El modelo procesa datos constantemente según se le provean en intervalos de baja magnitud, lo cual puede ahorrar tiempo y memoria en los sistemas que se dediquen a entrenar al modelo.

- **Adaptación en tiempo real:**
  Estos modelos pueden adaptarse a tiempo en tiempo real, en el caso de uso visto en los casos de fraudes bancarios, el modelo podría incrementalmente entrenar utilizando las transacciones del mes, de la semana o del día, según las necesidades del desarrollo.

Este enfoque también tiene sus limitaciones y vulnerabilidades a tomar en cuenta, entre ellas, el “olvido catastrófico”, en el cual el modelo presenta una deficiencia en la retención de datos pasados al entrenarse con datos nuevos, lo cual puede presentarse en forma de una disminución de la precisión general del modelo. También, este enfoque puede llevar al modelo a tener una tendencia hacia el sobreajuste, ya que podría llegar cierto momento durante su entrenamiento en el que el modelo podría sobre ajustar sus parámetros basado en un cambio abrupto de datos que podrían llegar a no representar la distribución normal de los datos.

Vale la pena mencionar, dentro del mismo enfoque del entrenamiento incremental se pueden tomar ciertas variaciones según el contexto del problema a resolver. (Van de Ven, Tuytelaars, Tolias., 2022)

- Basado en tareas
  - Aprender secuencialmente a resolver una serie de tareas distintas.

    - Durante el entrenamiento, el modelo tiene claro cuál es la tarea a resolver.
    - Ej. Aprendizaje sobre tácticas en deportes, aprendizaje sobre cómo tocar instrumentos musicales.
- Basado en dominio
    - Aprender a resolver un mismo problema en diferentes contextos.
    - Durante el entrenamiento, el modelo no tiene identificadas las tareas, porque todas tienen el mismo resultado.
    - Ej. Identificación de objetos bajo diferentes condiciones de luz.
- Basado en clases
    - Discriminar entre clases observadas de manera incremental.
    - El algoritmo debe distinguir entre todas las clases.
    - Ej. Clasificación de especies de animales, Clasificación de objetos.

Para este proyecto se utilizara SVM el cual tiene muchas ventajas dentro del contexto de entrenamiento incremental: (Shilton, et al., 2005)

- Los algoritmos de SVM típicamente encuentran un compromiso para minimizar el riesgo estructural del modelo y prevención del sobreajuste.
- El clasificador resultante puede ser modificado para que se ajuste a las necesidades del desarrollo.

Adicionalmente, se utilizará random forest el cual tiene como principales ventajas: (Hu, et al., 2018)

- Puede lograr reconocer patrones de actividad.
- Flexibilidad al momento de agregar ramas para el entrenamiento incremental.
- Rendimiento eficiente para hacer reconocimiento de patrones con un flujo constante de datos.

Con todo lo anteriormente expuesto, se puede observar que el entrenamiento incremental es una gran oportunidad para el desarrollo de la inteligencia artificial, abre la posibilidad del mejoramiento continuo de modelos existentes para poder alcanzar una precisión extremadamente alta y ganando experiencia con cada interacción que el modelo tiene con nuevos datos.

Descripción de implementación

Para la implementación del modelo de detección de fraude de tarjetas de crédito se utilizó un dataset simulado de transacciones con tarjetas de crédito legítimas y fraudulentas, que contiene registros desde el 1 de enero de 2019 hasta el 31 de diciembre de 2020. Este conjunto de datos tiene tarjetas de créditos de 1000 clientes que realizaron transacciones de 800 comercios.

Luego de un análisis exploratorio de las variables encontradas en el conjunto de datos, se modificó de tal manera que se pudiera maximizar el rendimiento del modelo, las variables del conjunto utilizado fueron las siguientes:

- cc_num:
  - Número de la tarjeta de crédito
- merchant:
  - Nombre del comercio
- category:
  - Categoría de comercio
- amt:
  - Cantidad de dinero gastado en transacción
- lat:
  - Latitud de ubicación del usuario
- long:
  - Longitud de ubicación del usuario
- city_pop:
  - Población de la ciudad del usuario
- merch_lat:
  - Latitud de ubicación del comercio
- merch_long:
  - Longitud de ubicación del comercio
- is_fraud:
  - Variable objetivo, está definida como booleano, es true si la transacción es fraudulenta y es false si la transacción es legítima
- amt_month:
  - Cantidad de dinero gastado en transacciones mensualmente
- amt_year:
  - Cantidad de dinero gastado en transacciones anualmente
- amt_month_shopping_net_spend:
  - Cantidad total de dinero gastado en transacciones
- first_time_at_merchant:
  - Booleano, es true si es la primera transacción del usuario con ese comerciante, es false si ya ha tenido transacciones en el pasado.
- dist_between_client_and_merch:
  - Distancia entre la ubicación del cliente y el comerciante, calculado con las coordenadas de longitud y latitud.
- trans_month:
  - Mes de la transacción
- trans_day:
  - Dia de la transacción
- hour:
  - Hora de la transacción

- year:
    - Año de la transacción
- times_shopped_at_merchant:
    - Cantidad total de transacciones entre el usuario y comerciante
- times_shopped_at_merchant_year:
    - Cantidad de transacciones entre el usuario y comerciante por año
- times_shopped_at_merchant_month:
    - Cantidad de transacciones entre el usuario y comerciante por mes
- times_shopped_at_merchant_day:
    - Cantidad de transacciones entre el usuario y comerciante por dia

Para el entrenamiento del modelo se dividió el conjunto de datos en 70% como conjunto de entrenamiento y 30% como conjunto de pruebas. El conjunto en general se dividirá en transacciones del año 2019 y año 2020, de esta manera el primer entrenamiento se realizará con transacciones del año 2019 y luego el segundo entrenamiento de manera incremental se realizará con las del año 2020.

Vale la pena mencionar que el desbalance extremo en el conjunto de datos entre transacciones legítimas y fraudulentas es de 99.47% a 0.52% respectivamente, para lograr disminuir el impacto negativo que esto podría llegar a tener se utilizó la técnica de sobre muestreo SMOTE (Synthetic Minority Over-sampling Technique), el cual genera nuevas muestras sintéticas de la clase minoritaria, es decir de las transacciones fraudulentas.

El proceso de SMOTE implica lo siguiente:

1. Seleccionar un ejemplo de las transacciones fraudulentas
2. Encontrar sus vecinos más cercanos que sean transacciones fraudulentas utilizando algoritmos como K-Nearest Neighbors (KNN).
3. Selecciona aleatoriamente uno de esos vecinos y calcula la diferencia entre ellos.
4. Obtiene el producto de esta diferencia por un número aleatorio entre 0 y 1, y agrega el resultado al ejemplo original para crear una nueva muestra sintética de la transacción.
5. Repite hasta poder alcanzar un nivel menor de desbalance entre los dos tipos de transacciones.

El modelo incremental de random forest utilizó los siguientes parámetros para su entrenamiento:

El modelo utiliza inicialmente 10 árboles de decisión para el entrenamiento inicial, luego en el segundo entrenamiento se le agregan 10 más para un total de 20 árboles de decisión en el modelo.
La profundidad máxima de cada árbol se limitó a 2 y el número mínimo de muestras requeridas para dividir un nodo interno se estableció en 5.
El parámetro del modo calentamiento se establece como verdadero, este es clave para poder entrenar de manera incremental el modelo.

El modelo incremental de SVM utilizó los siguientes parámetros para su entrenamiento:

Como clasificador se utiliza SGD, el cual permite hacer entrenamiento incremental mientras se mantienen las características claves de un SVM normal, se utilizó el parámetro de función de pérdida “hinge” debido a que este es un problema de clasificación binario.
Se aplica regularización L2 para evitar el sobreajuste por medio del parámetro de penalización.
En este caso el propio clasificador tiene habilitado la habilidad de hacer entrenamiento incremental simplemente utilizando la función partial_fit, la cual hace un entrenamiento incremental con datos nuevos, utilizando lo previamente aprendido por el modelo, sin necesidad de especificar el inicio del modo de calentamiento.

Análisis de resultados

Para el modelo que utiliza Random Forest se obtuvieron los siguientes resultados:

- ROC AUC Score:
    - Sobre los datos de 2019, el área bajo la curva es de 88.88%, lo que indica que el modelo tiene capacidad alta para distinguir entre las transacciones fraudulentas y legítimas. En 2020, este puntaje aumentó ligeramente a 89.62%, lo que sugiere una mejora en la capacidad de clasificación del modelo después del entrenamiento incremental. Sin embargo, este incremento es mínimo, por lo que quizás para futuras iteraciones se debería de considerar alimentarlo con más datos para mejorar este aspecto del modelo.
- Recall Score:
    - En 2019, el recall es del 84.28%, esta métrica representa la tasa de precisión en la que el modelo identifica correctamente las transferencias ilegítimas. Luego del entrenamiento incremental, utilizando los datos del 2020, se observó una ligera disminución en el puntaje con 83.89%, sin embargo, este puntaje sigue siendo bastante alto, por lo que sugiere que el modelo sigue siendo bastante efectivo en la identificación de las transferencias ilegítimas. Como posible punto de mejora en este aspecto, de manera similar al AUC Score, se podría considerar utilizar una mayor cantidad de datos o hasta incluso una división más granular en los entrenamientos incrementales, es decir en lugar de realizarlo por medio de incrementos anuales, se podrían realizar de manera semestral o trimestral para observar alguna mejora
- F1 Score:
    - El puntaje F1 que es una combinación de ambas métricas anteriores nos muestra que en general hay una ligera mejora en el rendimiento del modelo de 88.34% a 88.99% con la data de 2019 y 2020 respectivamente, lo que demuestra que si se puede lograr una mejora general del rendimiento del modelo utilizando el método de entrenamiento incremental.

Para el modelo que utiliza SVM se obtuvieron los siguientes resultados:

- ROC AUC Score:
    - En 2019, el área bajo la curva es de 89.05%, lo que muestra que el modelo tiene una alta capacidad para la distinción de ambos casos de transacciones. En 2020, se observa una mejora marginal, 89.39% después del entrenamiento incremental. Esto sugiere que aún hay margen de mejora con este modelo, quizás empleando otras características en el conjunto de datos para que el modelo pueda más fácilmente realizar la distinción de clases de transacciones.
- Recall Score:
    - Para el primer entrenamiento con los datos de 2019, el recall es del 83.16%, lo que indica que el modelo identificó correctamente el 83.16% de las transferencias fraudulentas. De manera similar al AUC Score, el recall aumentó ligeramente al 83.55% tras el entrenamiento incremental, por lo que aún hay margen de mejora para este aspecto.
- F1 Score:
    - En el puntaje general del modelo, se refleja lo anteriormente visto en los puntajes anteriores, tras el entrenamiento incremental se observa una mejora marginal en el modelo, 88.37% y 88.73% para 2019 y 2020 respectivamente.

A través de los resultados de ambos modelos podemos observar que cada uno pasó por una ligera mejora tras realizar el entrenamiento incremental, sin embargo, valdría la pena en futuras iteraciones observar cómo se comporta con diferentes variables en el conjunto de datos, con diferentes algoritmos y metodologías de machine learning e incluso con la inclusión de más datos, para observar si estas mejoras marginales observadas en ambos modelos puede aumentar para poder crear un modelo aún más robusto.

Metodología propuesta

Decidir entre reentrenamiento total o incremental de modelos dentro del contexto del proyecto puede llegar a depender de varios factores, como la disponibilidad de datos actualizados, la frecuencia de cambio en los patrones de tácticas para transacciones fraudulentas y los recursos computacionales disponibles para realizar el entrenamiento. Para dar una posible respuesta hacia la toma de esta decisión se desarrolló la siguiente metodología la cual evalúa algunos de los aspectos más importantes a considerar para elegir entre reentrenamiento total o entrenamiento incremental tomando en cuenta sus ventajas y desventajas.

Para ambos modelos se decidió realizar entrenamientos incrementales por cada mes en el dataset, es decir que habrá 24 entrenamientos en total para cada uno de los modelos, para de esta manera poder visualizar si los modelos tienden a tener alguna de las limitaciones observadas al momento de toparse ante data drifting.

Para el modelo de random forest se obtuvieron los siguientes resultados de entrenamiento.

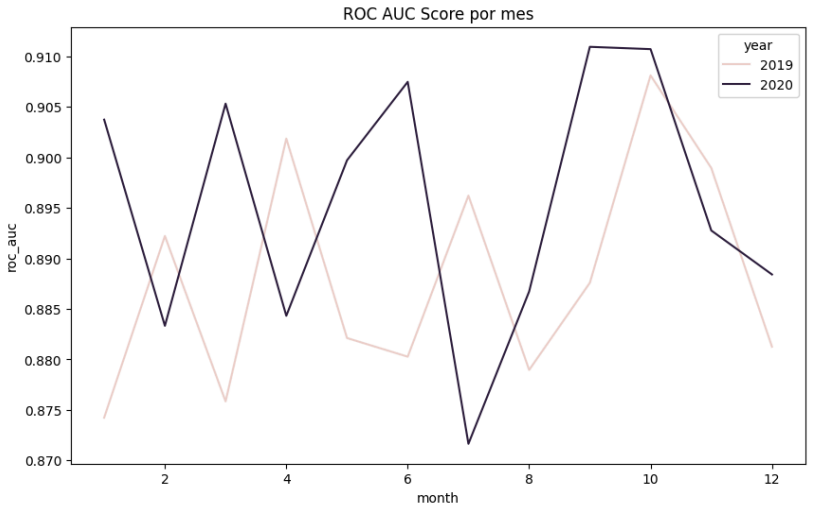

Fig. 1. ROC AUC Score - Random Forest

Fig. 2. Recall Score - Random Forest

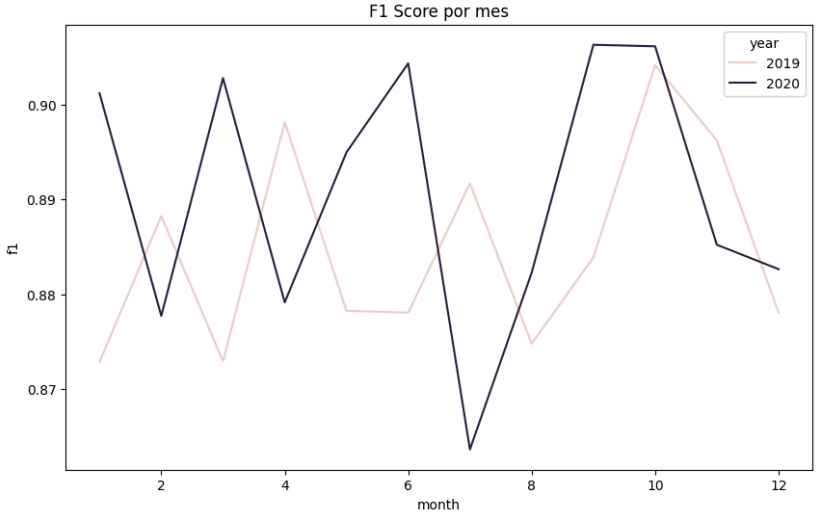

Fig. 3. F1 Score - Random Forest

Como se puede observar en las figuras mostradas, hay varios meses en los que el rendimiento del modelo baja, se puede destacar que en julio hay una disminución en el rendimiento general del modelo, en las métricas del 2020 hay más variabilidad en el modelo, lo que puede indicar que está siendo afectada por el data drifting y las variaciones que se encuentran en el conjunto de datos.

Esto indica que un reentrenamiento total podría ser adecuado para mantener un rendimiento estable, en especial cuando hay una variación en el modelo, debido a la inestabilidad encontrada de mes a mes se podría recomendar realizar un reentrenamiento semestral, de tal manera que al pasar 6 meses se vaya agregando la data recopilada en ese periodo al conjunto de datos total para poder hacer un entrenamiento efectivo y que el modelo pueda adaptarse a la variabilidad que tiene la data.

Se logró observar un alto porcentaje de predicciones correctas según lo observado en las matrices de confusión pertenecientes a ambos años, por lo que de esta forma se puede respaldar que se ha obtenido un modelo bastante efectivo en la detección de fraude.

258014	17752
46641	229371

Tabla 1. Matriz de confusión 2019

264565	12361
47344	229598

Tabla 2. Matriz de confusión 2020

Para el modelo incremental de SVM se obtuvieron las siguientes métricas:

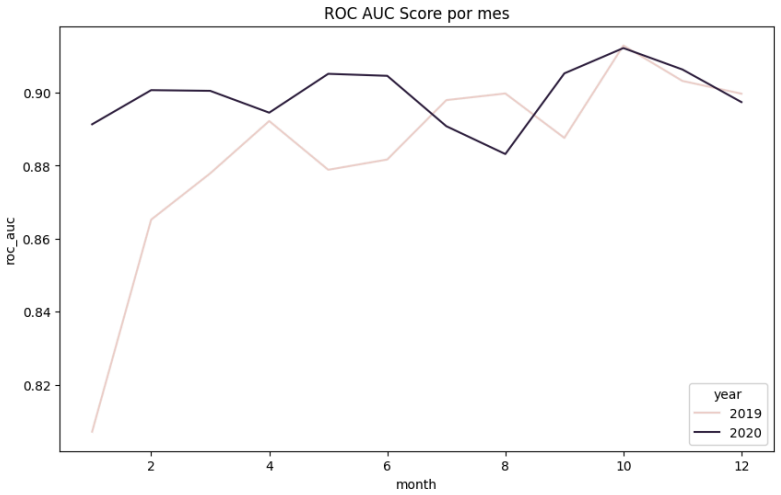

Fig. 4. ROC AUC Score - SVM

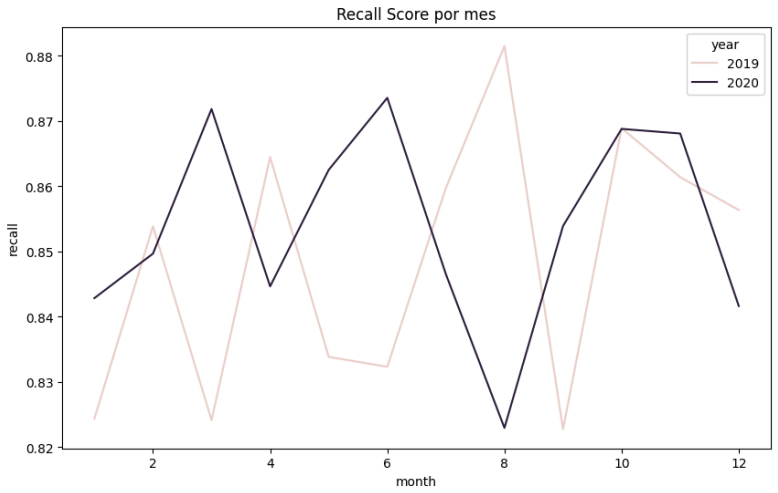

Fig. 5. Recall Score - SVM

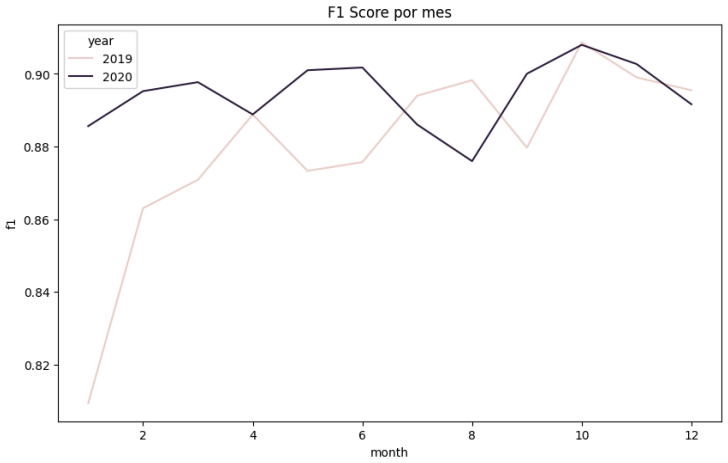

Fig. 6. F1 Score - SVM

Se logró observar que, de manera similar con el modelo Random Forest hay variaciones en el rendimiento, en espacial en abril, julio, agosto y diciembre, sin embargo, vale la pena mencionar que el rendimiento se mantiene mas estable durante mas tiempo que con el modelo de Random Forest, por esta razón un reentrenamiento total se podría considerar de manera anual para adaptarse a la variabilidad que pueda tener año con año el historial de transacciones, de esta manera se podrían obtener mejores métricas y resultados que reflejen un rendimiento más estable de parte del modelo.

El rendimiento observado en las matrices de confusión, de manera similar al modelo de Random Forest, refleja un buen rendimiento por parte del modelo, ya que tiene una tasa de predicciones positivas bastante alta y una tasa de falsos positivos baja en comparación, por lo que se podría considerar como un modelo exitoso.

261233	14533
47123	228889

Tabla 3. Matriz de confusión 2019

263366	13560
46124	230818

Tabla 4. Matriz de confusión 2020

Conclusiones

- Se deben de utilizar las estrategias adecuadas de entrenamiento según el contexto en el que se encuentre el proyecto y sus necesidades correspondientes.
- Se deben de manejar de manera adecuada los datos desequilibrados para no afectar el entrenamiento y poder mantener un buen rendimiento durante la evaluación de datos reales.
- La ingeniería de características es un aspecto importante a tener en cuenta cuando se desea obtener una mejor correlación entre las diferentes variables dentro de un conjunto de datos y por lo tanto mejorar el rendimiento general del modelo.
- El entrenamiento incremental es una metodología muy importante y util cuando se tienen datos con una alta variabilidad y poca estabilidad a lo largo del tiempo.

Recomendaciones

Algunas de las recomendaciones para futuras iteraciones tomando en cuenta lo visto durante el desarrollo del proyecto pueden ser:
En primer lugar, contar con conjuntos de datos que amplíen el plazo de tiempo en el que se tomaron las transacciones, es decir que tengas de años pasados y posteriores para de esta manera poder reflejar una diversidad mayor en las transacciones.
Asimismo, la selección de nuevas características dentro del conjunto de datos existente, para de esta manera poder garantizar que el modelo pueda relacionarse de mejor manera con los datos y poder obtener un mejor rendimiento en sus predicciones.
Otra estrategia que podría resultar clave para el mejoramiento del proyecto en general es el uso de otros modelos como Naive Bayes o redes neuronales para observar cómo se comportan y tener un mayor panorama en cuanto a cuál podría ser la mejor opción si en algún momento se desea ampliar el alcance del proyecto.

Referencias bibliográficas

Awan, A., (2023) What is Incremental Learning? https://www.datacamp.com/blog/what-is-incremental-learning

Shilton, A., Palaniswami, M., Ralph, D., & Tsoi, A. C. (2005). Incremental training of support vector machines. IEEE transactions on neural networks, 16(1), 114-131.

Domeniconi, C., & Gunopulos, D. (2001, November). Incremental support vector machine construction. In Proceedings 2001 ieee international conference on data mining (pp. 589-592). IEEE.

Hu, C., Chen, Y., Hu, L., & Peng, X. (2018). A novel random forests based class incremental learning method for activity recognition. Pattern Recognition, 78, 277-290.

Van de Ven, G. M., Tuytelaars, T., & Tolias, A. S. (2022). Three types of incremental learning. Nature Machine Intelligence, 4(12), 1185-1197.