25 nov 2025

Análisis Matemático (6/8) - a241b25

Invitados merrazquin@fi.uba.ar Análisis Matemático - CEIA 24Co2025

Archivos adjuntos Análisis Matemático (6/8) - a241b25
Análisis Matemático (6/8) - a241b25 - 2025/11/25 18:54 GMT-03:00 - Recording
Análisis Matemático (6/8) - a241b25 - 2025/11/25 18:54 GMT-03:00 - Chat

Registros de la reunión Transcripción Grabación

Resumen

Ise posgrados, Cristian A. Aballay, Gabriel Quiroga, Alejandro Valle, Cesar Orellana, Federico G. Zacchigna y Gabriela Sol Salazar discutieron sobre la minimización de la pérdida a través del concepto de Descenso de Gradiente, la diferencia entre parámetros (entrenables) e hiperparámetros (fijos) y la minimización del riesgo empírico. Se abordó la dificultad de la Clase 6, el concepto de optimización, el uso de diferentes funciones de pérdida para entrenamiento y prueba, la taxonomía de optimización, la intuición del *Gradient Descent* y el rol del *learning rate* como hiperparámetro. Finalmente, se exploró el impacto del *batch size* en la estabilidad del gradiente, con *mini batches* siendo un compromiso entre estabilidad y eficiencia computacional, y se aclaró que las preguntas técnicas en el código del TP son para autoevaluación.

Detalles

Longitud de las notas: Estándar

- **Introducción y Revisión de Material** La reunión comenzó con un saludo y la invitación a realizar consultas sobre el Trabajo Práctico (TP). Gabriel Quiroga comentó que estaba leyendo el material para entender el clasificador Valan antes de empezar el TP, lo que lleva tiempo. Ise posgrados preguntó por las impresiones sobre los videos de la semana, a lo que Cristian A. Aballay y

Gabriel Quiroga respondieron que eran entendibles, claros y más cortos en comparación con la clase anterior (00:00:00).

- **Temas y Dificultad de las Clases** Ise posgrados aclaró que los videos de la semana eran menos que la clase anterior (dos versus cuatro), y que la lista de reproducción de YouTube sobre Back Propagation es optativa, explicando de dónde salen las fórmulas. Ise posgrados también expresó que, subjetivamente, considera esta clase (clase 6) la más difícil de la materia, ya que "ata muchos cabos" (00:03:38). Además, mencionó una recomendación de un alumno de un curso anterior que sugería ver los videos de la clase 5 y 6 en un *loop* para comprender cómo se conectan las piezas del rompecabezas (obtener las derivadas de la pérdida y usarlas) (00:05:27).
- **Concepto de Optimización y Minimización** Ise posgrados explicó que en machine learning la optimización es distinta a la optimización común, y por convención, el enfoque siempre será la minimización de un campo escalar (00:05:27). Mencionó que maximizar la verosimilitud se convierte en minimizar la pérdida a través de la *Negative Log Likelihood* (NLL). El problema principal es que en optimización real se necesita la verdadera distribución de los datos, pero como solo se tienen observaciones con posible ruido, solo se tiene una estimación del error, lo que requiere separar el *dataset* en entrenamiento y prueba (00:06:56).
- **Minimización del Riesgo Empírico** El esquema general del aprendizaje supervisado busca minimizar una función de pérdida, *J* de *Tita*, que depende de los parámetros del modelo (00:08:38). Ise posgrados explicó que se utiliza el *framework* de la minimización del riesgo empírico, que aproxima la esperanza de la pérdida mediante el promedio sobre todos los elementos del *dataset* (00:10:03). Alejandro Valle consultó sobre el término de esperanza, a lo que Ise posgrados clarificó que, en términos probabilísticos, es una medida de centralidad, o un "promedio sobre toda la población infinita de todos los posibles valores" (00:11:19).
- **Parámetros vs. Hiperparámetros** Alejandro Valle preguntó sobre la diferencia entre *Titas* y los hiperparámetros (00:12:40). Ise posgrados explicó que *Tita* simboliza el conjunto de parámetros del modelo (como los betas en regresión lineal) que se entrenan, mientras que los hiperparámetros son valores que se fijan antes del entrenamiento (como el grado del polinomio o la temperatura en *simulated annealing*) (00:14:11).
- **Funciones de Pérdida en Entrenamiento y Prueba** Ise posgrados enfatizó que el objetivo es minimizar la pérdida en el conjunto de prueba (*test*), y la minimización en el conjunto de entrenamiento (*train*) es una herramienta para

lograrlo (00:15:33). Señaló que las funciones de pérdida para *train* y *test* no necesariamente tienen que ser las mismas (00:17:13). Una función de pérdida en *test* (como la *accuracy*) puede ser fácil de entender para el negocio, pero no óptima para el modelo, mientras que una función en *train* (como la *cross-entropy*) debe ser derivable, lo que es clave para el *backpropagation* (00:18:40). Alejandro Valle expresó confusión sobre si el uso de diferentes funciones de pérdida era una forma de "hacer trampa", a lo que Ise posgrados aclaró que la función de *test* se define por la necesidad de negocio para evaluar el modelo, y la función de *train* se diseña para que, al minimizarla, también se minimice la de *test*, pero con comodidades asociadas (00:20:59).

- **Taxonomía de Optimización y Restricciones** Se presentó la taxonomía de los problemas de optimización, indicando que se centrarán en la optimización continua y, en las próximas clases, en la optimización sin restricciones. Ise posgrados señaló que a la industria no le gusta la optimización con restricciones, ya que introduce un "conflicto de intereses" matemático donde no todos los valores de *Tita* son válidos (00:23:49). Cristian A. Aballay solicitó un ejemplo de optimización con restricciones, y Ise posgrados proporcionó un caso de regresión lineal donde los pesos deben ser mayores o iguales a cero, lo cual es una restricción lógica de negocio (00:26:21).
- **Grado de Certidumbre en Optimización** Ise posgrados discutió que en problemas de optimización, el único caso trivial es aquel donde se conoce el mínimo global (00:27:45). Para todos los demás casos, especialmente en modelos complejos como redes neuronales, el optimizador solo puede garantizar que ha alcanzado un mínimo local, y nunca se puede saber si existe otro valor de pesos que podría dar un mejor resultado (00:29:07).
- **La Intuición del Gradiente Descendente (*Gradient Descent*)** La familia de *Gradient Descent* establece la base para la optimización (00:30:25). La idea es alterar los valores de *Tita* para reducir *J* de *Tita*, lo cual requiere que la función de costo sea derivable respecto a *Tita*. El algoritmo se basa en que el cambio en *Tita* es proporcional a la derivada de *J* respecto a *Tita* multiplicada por -1, ya que el gradiente indica la dirección de máximo crecimiento y, por lo tanto, la dirección opuesta indica la máxima disminución (00:31:47).
- **El *Learning Rate* (Tasa de Aprendizaje)** Cesar Orellana preguntó si el factor de proporcionalidad (alfa o lambda) en la ecuación de *Gradient Descent* es un hiperparámetro, y cómo se determina qué tanto bajar. Ise posgrados confirmó que sí es un hiperparámetro, conocido como el *step size* o *learning rate*, y si se conociera su valor óptimo, se resolvería el problema de convergencia en un solo paso (00:34:48). El valor tiene que ser lo suficientemente grande para avanzar, pero lo suficientemente pequeño para mantenerse dentro de un

entorno local. Alejandro Valle concluyó que su selección es más que todo "prueba y error", aunque Ise posgrados mencionó reglas empíricas para rangos iniciales, como 10 a la -2 a 10 a la -6 para redes neuronales usando *Adam* (00:35:55).

- **Métodos de Primer y Segundo Orden** Ise posgrados explicó la diferencia entre métodos de primer orden (usan el gradiente) y de segundo orden (usan el gradiente y el Hessiano), en referencia al grado de las derivadas utilizadas (00:37:34) (00:40:27). Para modelos grandes, el cálculo de derivadas de orden superior (Hessiano, que es una matriz *n* por *n*, o tensores de orden superior) se vuelve computacionalmente inviable debido al tamaño, lo que hace que los métodos de primer orden sean más comunes (00:39:09). Aunque los métodos de segundo orden tienen una tasa de convergencia cuadrática (más rápida en iteraciones), los de primer orden son linealmente más rápidos y terminan convergiendo en menos tiempo de reloj (00:44:29).
- **Tensores en *Deep Learning*** Alejandro Valle preguntó por qué se "huye" de los tensores, dado su uso en *deep learning* (00:40:27). Ise posgrados explicó que en *deep learning*, los tensores se usan a menudo como matrices apiladas para paralelizar cálculos matriciales (como multiplicar varias matrices por el mismo vector) (00:41:37). Las operaciones no suelen requerir el "álgebra tensorial" complejo y extraño de los tensores en su definición matemática completa (00:42:59).
- **Definición y Propiedades de *Gradient Descent*** Ise posgrados formalizó la definición de *Gradient Descent*: el nuevo valor de *Tita* es el valor actual menos el *learning rate* multiplicado por el gradiente de *J*, asegurando que el paso se dé en la dirección de máximo decrecimiento (00:45:46). Las propiedades del método incluyen que el *learning rate* típicamente es una sucesión (no un valor fijo) que converge al mínimo local. La inicialización de los parámetros generalmente se hace con valores aleatorios (00:47:26).
- **Estrategias para el *Learning Rate* (Schedulers)** Una familia de ideas para el *learning rate* es el *scheduler*, que propone iniciar con un valor de *gamma* (tasa de aprendizaje) alto porque hay poco que perder, y luego disminuirlo a medida que avanzan las iteraciones (decreciente respecto a *T*), para evitar rebotar alrededor del mínimo local (00:47:26). Ise posgrados mostró ejemplos de *schedulers* de la librería PyTorch, que son muy comunes (00:50:16).
- **Estimación del Gradiente y *Batch Size*** Ise posgrados introdujo el *batch size* (*m*), que es la cantidad de observaciones utilizada para estimar el gradiente de *J* (00:52:06). La pregunta es, si *m* está entre 1 y *n* (el tamaño del *dataset*), ¿cuál es el tamaño óptimo? El deseo es usar un *m* lo más pequeño posible, ya

que el *backwards* es lo que realmente entrena el modelo (00:53:51). El problema con un *m* muy pequeño es la alta variabilidad de la estimación del gradiente, aunque siga siendo un estimador insesgado (00:55:08).

- **Eficiencia de la Muestra y Computación Paralela** Ise posgrados explicó que para reducir 10 veces el error estándar de la estimación se necesitan 100 veces más observaciones (por la raíz cuadrada del tamaño de la muestra), lo que sugiere que aumentar *m* no es costo-eficiente en tiempo. Sin embargo, el procesamiento paralelo en *hardware* (GPU, TPU) permite procesar muchas entradas al mismo tiempo, lo que rompe el costo secuencial y hace que un *batch size* mayor sea prácticamente gratis en tiempo hasta que la memoria del acelerador se agota (00:56:26). Finalmente, Ise posgrados presentó los tres enfoques de *batch size*: *stochastic* (m=1), *full batch* (m=n) y *mini batches* (m es mucho menor que n pero mayor que 1) (00:57:43).
- **Conceptos Fundamentales de Descenso de Gradiente** Ise posgrados recapituló el proceso de Descenso de Gradiente para redes diferenciables, que implica inicializar los parámetros θ, seleccionar m observaciones (tamaño de *batch*), generar predicciones, calcular la función de pérdida y su derivada (*backpropagation*), y promediar para obtener los gradientes de J respecto a cada θ (00:59:24). Luego, se aplica el delta θ basado en un método definido, como la regla de *gradient descent*, y el proceso se repite hasta alcanzar un criterio de corte (01:00:47).
- **Estimación del Gradiente con Mini-Batches** Gabriel Quiroga preguntó sobre la estimación del gradiente utilizando m observaciones, y Ise posgrados aclaró que estas son observaciones cualesquiera del conjunto de entrenamiento, no necesariamente similares (01:02:13). Ise posgrados explicó que promediar los gradientes de m observaciones (mini-batch) resulta en una mejor estimación y reduce el ruido en comparación con usar una sola observación (estocástico) (01:03:39). Usar un *mini batch* diferente en cada iteración es lo recomendable para aprovechar la diversidad del conjunto de datos (01:05:27).
- **Visualización del Ruido en la Estimación del Gradiente** Ise posgrados ilustró que las curvas de nivel de los valores de J respecto a θ muestran que la estimación estocástica del gradiente es muy ruidosa (01:03:39). Se mostró que el uso de mini-batches mitiga el ruido, y el uso del *batch* completo ofrece la máxima estabilidad, aunque a expensas de un mayor tiempo de procesamiento (01:05:27). La noción es que la variación del *batch size* influye en la estabilidad y el ruido de la estimación del gradiente, lo que Federico G. Zacchigna identificó como un posible regularizador para salir de mínimos locales (01:37:32) (01:42:07).

- **Modelo de Regresión Logística y su Derivada** lse posgrados describió la regresión logística como un modelo simple para clasificación binaria, similar a una neurona con función de activación sigmoidea (01:20:47). La derivada de la función sigmoide se calcula de manera eficiente como el valor tomado multiplicado por uno menos el valor, lo que simplifica el cálculo de la derivada (01:22:13). Para el entrenamiento se utiliza la *cross entropy*, en parte porque sus términos se cancelan al aplicar la regla de la cadena, resultando en un gradiente simple de $\hat{Y} - Y$ (01:23:46).
- **Implementación del Descenso de Gradiente para Regresión Logística** Se presentó una implementación en código que demostró el entrenamiento de una regresión logística utilizando un *dataset* de cáncer de mama (01:27:46). lse posgrados detalló los pasos para *forward* y *backward* propagation, incluyendo la inicialización de pesos y el cálculo de las derivadas para el *batch size* mayor a uno (01:26:32) (01:33:07). Se observó que el modelo aprende, mostrando una disminución de la pérdida en el conjunto de prueba, y que entrenar minimizando *cross entropy* mejora métricas como la *accuracy* (01:34:28).
- **Impacto del Tamaño del Batch (*Batch Size*) en el Entrenamiento** Se analizó el efecto del *batch size* en la variabilidad de la pérdida en la prueba, mostrando que un *batch size* de 1 produce curvas más ruidosas debido a la estimación del gradiente con una sola observación (01:34:28). Al aumentar el *batch size* a 10, las curvas se vuelven más suaves y similares, lo que confirma que promediar más observaciones reduce la variación (01:37:32). Un *batch size* de 100 resultó en curvas extremadamente suaves y pegadas, con muy poca variabilidad, aunque esto conlleva un mayor costo computacional en términos de tiempo reloj si no se aprovecha el procesamiento paralelo (01:39:03).
- **Consideraciones sobre el Tiempo de Procesamiento** lse posgrados enfatizó que, aunque un *batch size* mayor ofrece mayor estabilidad, el tiempo de procesamiento puede aumentar si el tiempo reloj se correlaciona con la cantidad de *forwards* (01:39:03). Esto implica que un *batch size* intermedio, como $m=10$, puede ser preferible a un *batch size* muy grande si el tiempo de cálculo es secuencial. La estabilidad es deseable solo si el tiempo de iteración no se ve afectado negativamente (01:40:34).
- **Consideraciones Prácticas y *Hardware* para la Elección del Batch Size** Federico G. Zacchigna y Gabriel Quiroga consultaron sobre el papel del *batch size* como regularizador y su dependencia del *hardware*. lse posgrados confirmó que una menor variabilidad puede ayudar a salir de mínimos locales y que el *hardware* es crucial. Una heurística es usar potencias de dos para el *batch size* y aumentarlo hasta que la GPU se quede sin memoria (Out of

Memory) (01:42:07) (01:44:53). La dificultad de predecir el *batch size* óptimo se debe a múltiples factores, como el formato de representación de los pesos (e.g., FP32 vs. BF16) y la necesidad de almacenar gradientes y salidas de capa, lo que hace que la experimentación sea más práctica que el cálculo teórico (01:46:14) (01:48:57).

- **Aclaración sobre Preguntas Técnicas en el TP** Gabriela Sol Salazar preguntó sobre unas preguntas técnicas (etiquetadas Q) encontradas en el código del Trabajo Práctico (TP) que no eran parte de la consigna (01:48:57). Ise posgrados aclaró que estas preguntas son para autoevaluación (un *sanity check* para los estudiantes) y que no forman parte de lo que deben entregar, además de que sus respuestas están disponibles en el campus (01:50:43).

Pasos siguientes recomendados

No se encontraron próximos pasos sugeridos para esta reunión.

Revisa las notas de Gemini para asegurarte de que sean correctas. Obtén consejos y descubre cómo toma notas Gemini

Danos tu opinión sobre el uso de Gemini para tomar notas en una breve encuesta.