Laboratorio 4: SGD & Redes Neuronales

Inteligencia Artificial - CC3085

José Antonio Mérida Castejón

11 de febrero de 2026

Task 1 - Teoría

Responda con criterio y anàlisis de ingeniero. No se esperan definiciones de libro, si no que analice las consecuencias de las decisiones de diseño.

El Dilema del "Step Size"

Durante la clase hablamos sobre el tamaño del paso. Suponga que está entrenando una red neuronal profunda con SGD (Estocástico) y nota que la función de pérdida (Loss) disminuye rápidamente al inicio, pero luego empieza a oscilar violentamente sin llegar nunca a un valor mínimo estable.

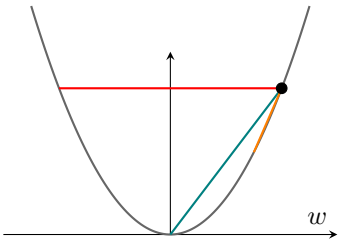
1. *Explique que está sucediendo geométricamente en la superficie de error*

   En términos simples, cada uno de nuestros 'saltos' nos está mandando hacia 'el otro lado' del mínimo que queremos encontrar.

   En términos más matemáticos, la gradiente nos indica la dirección dónde la función es más creciente (y su negativo, la dirección del descenso más pronunciado). Es decir, la información con la que contamos es únicamente una dirección, sin saber la distancia exacta a la que se encuentra el mínimo que buscamos. Esta distancia que se 'recorre' en cada iteración se determina únicamente a través del Learning Rate (tamaño del paso).

   Teniendo un Learning Rate demasiado alto, el tamaño del paso calculado supera la distancia real al mínimo. Esto provoca que nuestros pasos jamás sean tan precisos como para ubicar un mínimo, resultando en un rebote constante entre las "paredes"' de la función de error.

   A continuación, podemos visualizar este comportamiento en una superficie de error simple en 2 dimensiones:

   Dentro de la gráfica, la línea roja indica el comportamiento utilizando un step size alto. Podemos ver claramente que nos movemos del punto inicial en la dirección correcta (ya que

el mínimo se encuentra hacia la izquierda) pero simplemente llegamos hacia "el otro lado". Este comportamiento se repetiría, dónde estando hacia la izquierda nos moveríamos hacia la derecha con un tamaño de paso idéntico llegando al punto original. Por otro lado, la línea teal simboliza un learning rate "perfecto" iniciando desde nuestro mismo punto inicial. En este caso, nos dirijimos en la dirección correcta y además la longitud de paso matchea perfectamente. Por último, tenemos un learning rate bajo con la línea naranja. Podemos ver que nuestro paso se dirije en la dirección correcta pero claramente necesitaríamos más iteraciones para llegar al mínimo que deseamos.

En resumen, un Learning Rate demasiado alto causa que oscilemos entre diferentes puntos cercanos al mínimo pero sin poder acercarnos, "saltando por encima" del mínimo de la función error en lugar de descender hacia el fondo.

2. *Justifique por qué una estrategia de Learning Rate Decay ($\eta = \frac{1}{\sqrt{t}}$) solucionaría este problema mejor que simplemente elegir un learning rate constante muy pequeño desde el inicio.*

   Si utilizamos un Learning Rate demasiado pequeño, tenemos dos problemas principales que enfrentamos:

   - **Convergencia lenta:** Si nuestro punto de inicio está lejos del mínimo (imaginemos la parte alta de la curva del inciso anterior), tomar pasos diminutos significa que necesitaríamos una gran cantidad de iteraciones para llegar al fondo. Básicamente, estaríamos invirtiendo recursos computacionales en un aprendizaje demasiado lento.
   - **Mínimos locales:** Al dar pasos tan pequeños, perdemos la capacidad de "saltar" fuera de pequeños valles o imperfecciones de la superficie. Es muy probable que el modelo se estanque en la primera solución (buena o mala) que encuentre en lugar de buscar el mínimo global real.

   Mientras que un Learning Rate demasiado grande presenta los problemas mencionados en el inciso anterior.

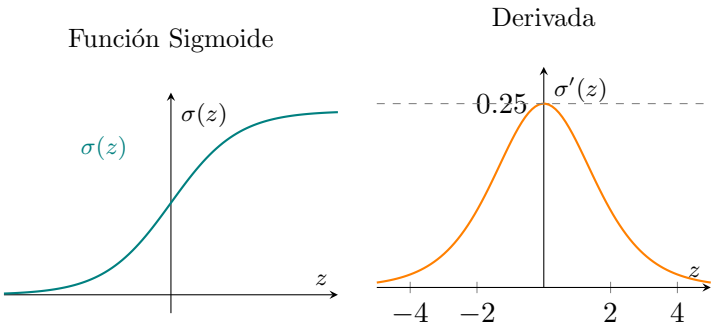
   La solución a esto sería utilizar Decay, aquí obtenemos un poco de lo mejor de ambos mundos. En las etapas iniciales, queremos que el modelo sea relativamente "agresivo" para llevarnos a la cercanía del mínimo global (o algún otro mínimo suficientemente cercano, pero ese es tema aparte) en pocas iteraciones con la capacidad de "escapar" de los mínimos locales. Mientras que, a mayor número de iteraciones buscamos dar pasos más pequeños y exactos para poder continuar mejorando el rendimiento y llegar a una solución más precisa.

Gradiente en Cero

Durante la clase hablamos sobre como evitar las gradientes cero. En base a eso responda

a) *Si usted diseña una red profunda utilizando únicamente la función Sigmoide en todas las capas ocultas, es muy probable que sufra el problema del Vanishing Gradient (Desvanecimiento del Gradiente). Explique matemáticamente por qué ocurre esto al hacer Backpropagation (piense en la derivada máxima de la sigmoide).*

   Basándonos en la forma de la función sigmoide, podemos darnos cuenta que su derivada tiene valores muy bajos. Observando la siguiente gráfica:



Podemos ver que el valor máximo de la derivada es **0.25**, dónde adicionalmente al acercarse a los extremos tiene valores súmamente bajos (zona de saturación). Adicionalmente, esto también es un problema ya que la función sigmoide "aprieta" los valores hacia el intervalo $(0, 1)$. Consecuentemente, es bastante probable que nuestros valores se encuentren dentro de nuestra zona de saturación.

Recordemos también que al utilizar la regla de la cadena durante *Backpropagation* para calcular el gradiente de las primeras capas, esencialmente estamos realizando una multiplicación continua de derivadas parciales:

$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial \hat{y}} \cdot \underbrace{\sigma'(z_n) \cdot w_n \cdot \cdots \cdot \sigma'(z_1)}_{\text{Multiplicación de términos} \leq 0{,}25} \cdot x$$

Si analizamos la ecuación, notamos el que estamos multiplicando repetidamente términos $\sigma'(z)$. Dado que ya observamos gráficamente que el *mejor* caso para estos términos es 0,25, estamos multiplicando fracciones pequeñas una y otra vez.

El resultado es que el gradiente decrece exponencialmente a medida que retrocedemos en las capas (*Vanishing Gradient*). Básicamente, la señal de error se diluye tanto que las primeras capas jamás reciben la información necesaria para actualizar sus pesos. Adicionalmente, las primeras capas de una red neuronal son clave al aprender características básicas (o features importantes en otras palabras).

b) ¿Por qué la función ReLU mitiga este problema en la parte positiva del eje?

La derivada de la función ReLU es 1 para valores positivos y 0 para valores negativos. Considerando únicamente valores positivos, la "señal" del error se mantiene intacta a lo largo del Backpropagation en comparación a la disminución exponencial utilizando la función sigmoide.

Capacidad del Modelo

Si comparamos una Red Neuronal con 1 capa oculta de 100 neuronas contra una Red Neuronal con 10 capas ocultas de 10 neuronas cada una (ambas tienen un número similar de parámetros).

a) ¿Cuál de las dos tiene mayor capacidad para modelar funciones no lineales complejas y composicionales? ¿Por qué? Base se respuesta con lo visto en clase ("¿Por qué redes profundas?").