

Universidad de Buenos Aires
Facultad de Ingeniería

Postgrado de Inteligencia Artificial
Guia de ejercicios y Trabajos Prácticos

CLASE 1

1. Resolver el problema de la XOR con 3 neuronas:
 - a. Implementar SGD para resolver la XOR.
 - b. ¿Cuántos parámetros desconocidos tiene el modelo?
 - c. ¿Cuáles son los hiper-parámetros del modelo? Explicar que pasa cuando se varía el learning rate.
 - d. Una vez entrenado el modelo, realizar predicciones para verificar el funcionamiento.

CLASE 2

1. Stochastic Gradient Descent para encontrar el máximo de una función

- a. Implementar en Python un algoritmo basado en Stochastic Gradient Descent para encontrar numéricamente el mínimo de $y = -(x - 2)^2 + 3$. ¿Cuál es el máximo valor que encontró el algoritmo y para qué valor de x? ¿En qué momento detuviste la búsqueda y por qué? ¿Cómo inicializamos los valores random del algoritmo y cómo afectan la convergencia? ¿Cuáles son los hiper parámetros del algoritmo?

2. Mini-Batch Gradient Descent para entrenar un modelo cuadrático

- a. Crear un dataset sintético con 10,000 muestras a partir de la siguiente expresión (donde “x” es la feature que toma valores entre 0 y 4, e “y” es la salida):

$$y = -(x - 2)^2 + 3 + 0.2 * \sin(16 * x)$$

- b. Suponer que se tiene un modelo, tal que la relación entre la salida y la entrada está dada por la ecuación presentada a continuación. Suponiendo que se va a utilizar mini-batch SGD para encontrar los valores óptimos de los pesos w1, w2 y w3 tal que minimizan el ECM (Error Cuadrático Medio) (el ECM es la función de costo para este problema), ¿cuál sería la regla de actualización de cada parámetro?. Implementar el algoritmo en Python (usar

mini-batch), encontrar los valores óptimos para w_1 , w_2 y w_3 y reportar la evolución del ECM con el número de epochs.

$$\hat{y} = w_1 x^2 + w_2 x + w_3$$

CLASE 3

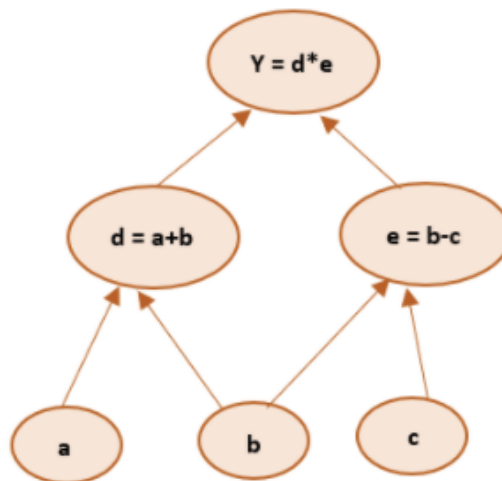
1. Mini-Batch Gradient Descent para entrenar un modelo cuadrático - PyTorch

- a. Implementar el ejercicio 2 de la Clase 2 utilizando PyTorch. Pueden tomar de referencia los notebooks vistos durante la clase.

CLASE 4

1. Backpropagation y regla de la cadena

- a. Automatic Differentiation: para el grafo de cómputo que se muestra en la siguiente imagen, con $a=1$, $b=2$ y $c=3$, calcular el valor final de Y y la derivada de Y respecto a cada una de las entradas (a , b y c). Para hacer el cálculo, primero realizar el paso forward y luego utilizar backpropagation con la regla de la cadena.



- b. Numerical Differentiation: para el grafo de cómputo que se muestra en la siguiente imagen, con $a=1$, $b=2$ y $c=3$, utilizar la siguiente expresión y un valor de h pequeño para calcular las derivadas respecto de cada entrada. Comparar los resultados con los obtenidos en el punto (a)

$$\frac{\partial Y}{\partial a} = \lim_{h \rightarrow 0} \frac{Y(a+h, b, c) - Y(a, b, c)}{h}$$