**Clase 3 – Combinador lineal**

**ADALINE – Adaptative Linear Element**

🡪 Red neuronal de una sola neurona

🡪 Adapta los pesos teniendo en cuenta el **error cometido al responder con respecto al valor esperado**

🡪 Algoritmo LMS o regla delta 🡪 Determinar pesos a fin de **minimizar error cuadratico medio**

Diagrama

Descripción generada automáticamente

**Entrenamiento del combinador lineal**

🡪 Método que permita calcular el vector W

**Derivación**

🡪 Rapídez con la que cambia el valor de la función

**Gradiente**

🡪 Maxima razón de cambio de la función

**Minimización de funciones usando el gradiente**

🡪 Función **continua**

1. Tomo punto dentro del dominio
2. **Calculo vector gradiente** en el punto
3. Sumo al punto anterior una **fracción del gradiente negativo** 🡪 Para ir hacia el **mínimo**
4. **Repetir hasta que la diferencia sea inferior a cierta cota**

**Minimización de funciones usando el gradiente estocastico**

**(COMPLETAR)**

**Clase 3B – Combinador lineal**

Diagrama, Dibujo de ingeniería

Descripción generada automáticamente**Combinador lineal**

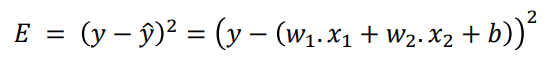
🡪 Resuelve un problema de **regresion lineal**

🡪 Se tiene una **función de error** 🡪 Error cuadratico medio

🡪 Tecnica de optimización 🡪 Descenso del gradiente estocastico

**Descenso del gradiente estocastico**

🡪 Se busca minimizar el error para un ejemplo especifico de la muestra



🡪 Calculamos el **gradiente** para saber **como modificar los pesos**

**Neurona general**

Diagrama

Descripción generada automáticamente

Patrón de fondo

Descripción generada automáticamente con confianza baja**Funciones derivables**

**Lineal** 🡪

Texto

Descripción generada automáticamente

**Sigmoide 🡪**

Imagen que contiene Patrón de fondo

Descripción generada automáticamente

**Sigmoide entre -1 y 1 🡪**

**Entrenamiento de neurona no lineal**

1. Elección de valor de alfa
2. Inicializar pesos W y b con valores random
3. Mientras la variación del ECM sea mayor a la cota:
   1. Para cada ejemplo:

🡪 Ingresar ejemplo a la red

🡪 Calcular el error

🡪 Actualizar pesos

Diagrama

Descripción generada automáticamente con confianza media

🡪 Actualizacion de pesos

Imagen que contiene Gráfico

Descripción generada automáticamente

🡪 Calculo del error

**Entropia cruzada binaria**

Imagen que contiene Diagrama

Descripción generada automáticamente🡪 **Función de costo** que puede usarse con **neuronas no lineales con función sigmoide** entre 0 y 1

🡪 y es el valor binario esperado

🡪 y sombrero es la salida de la neurona 🡪

**Clasificación con mas de 2 clases**

🡪 **Varias neuronas no lineales** para resolver problema de clasificación de mas de 2 clases

🡪 **Cada neurona de la capa de salida** buscara **responder por un valor de clase** distinto

🡪 El error de la capa será la **suma de los errores** de las neuronas

Texto

Descripción generada automáticamente

**Función softmax**

🡪 **Función de activación en la ultima capa**

🡪 **Normaliza** la salida de la red a una **distribución de probabilidad**

**Función de costo: Negative Log-Likelihood (NLL)**

Texto

Descripción generada automáticamente con confianza media

🡪 y es un **vector binario** que **vale 1 solo en la posición correspondiente al valor de clase esperado**

**🡪** s es la **neurona correspondiente** al valor de clase esperado

🡪 Solo se evalua en la **neurona correspondiente a la salida esperada**

🡪 Mientras **mas cerca de 1** 🡪 Menor error

🡪 Menor valor de la neurona softmax correspondiente a la clase correcta 🡪 Mayor error

Tabla

Descripción generada automáticamente