**Clase 4 – Multiperceptron**

Diagrama, Esquemático

Descripción generada automáticamente**Arquitectura**

**Redes multicapa**

🡪 Con 1 neurona no se pueden resolver ciertos problemas 🡪 Ej: XOR 🡪 No es linealmente separable

Gráfico, Gráfico de dispersión

Descripción generada automáticamente🡪 Se busca un **algoritmo mas general** que permita **integrar el aprendizaje entre las dos capas**

**Problema no separable linealmente**

🡪 Aplicar **descenso en la dir. del gradiente** **sobre la superficie de error** expresada como una **función de pesos**

🡪 Se tienen en cuenta los **pesos de los arcos que unen AMBAS capas**

🡪 Se conoce la **respuesta esperada a cada entrada** 🡪 Regla delta

**Algoritmo backpropagation**

🡪 Conjunto de vectores {(X1, Y1), … , (Xp, Yp)} 🡪 Corresponden a la función

Dibujo de una persona

Descripción generada automáticamente con confianza baja🡪 Se busca entrenar la red para que aprenda una aproximación

**Capa oculta**

🡪 Se tiene **en la capa oculta:**

* Un **ejemplo de entrada**
* **Entrada ne**ta de la **j-esima neurona de la capa oculta**
* **Salida** de la j**-esima neurona de la capa oculta**

🡪 Se tiene en la **capa de salida:**

* **Entrada neta** de la **k-esima neurona de la capa de salida**
* **Salida** de la **k-esima neurona de la capa de salida**

**Actualizacion de pesos**

🡪 Error en una sola unidad de la capa de salida

Texto

Descripción generada automáticamente con confianza media

* y: salida deseada
* o: salida real
* p: p-esimo vector de entrada
* k: k-esima unidad de salida

Imagen que contiene objeto, reloj

Descripción generada automáticamente

🡪 Se busca minimizar

Texto

Descripción generada automáticamente

🡪 Se tomará el valor negativo del gradiente

Texto

Descripción generada automáticamente

🡪 **Corrección** para los pesos de los arcos **entre la capa de entrada y la oculta**

Imagen de la pantalla de un celular de un mensaje en letras negras

Descripción generada automáticamente con confianza baja

**Entrenamiento**

1. Aplicar vector de entrada y calcular su salida
2. Calcular el error
3. Determinar **dirección (+ o -) de cambio** de los pesos para reducir el error
4. Determinar la cantidad precisa **para cambiar cada peso**
5. **Corregir los pesos** de las conexiones
6. Repetir hasta reducir el error

Tabla

Descripción generada automáticamente**Matriz de confusión**

🡪 Aciertos sobre la **diagonal**

**Precision:** Proporción de predicciones correctas **sobre una clase**

**Recall: proporción de ejemplos** de una clase que son **correctamente clasificados**

**Accuracy: Performance general** sobre todas las clases 🡪 Aciertos/Total de ejemplos

Diagrama, Texto

Descripción generada automáticamente

Tabla

Descripción generada automáticamente**Tipos de descenso de gradiente**

Gráfico, Gráfico de dispersión

Descripción generada automáticamente**Capacidad de generalización**

🡪 Red neuronal formada por una única neurona

🡪 Correcta clasificación

🡪 Red generalizada correctamente

**Sobreajuste**

🡪 Red neuronal que usa 2 ocultas

Gráfico, Gráfico de dispersión

Descripción generada automáticamente🡪 Cada hiperplano busca la **mayor proximidad a los ejemplos** 🡪 Algunos se **terminan clasificando mal**

Diagrama

Descripción generada automáticamente

**Reducción del sobreajuste**

* Incrementar la cant. de ejemplos
* Reducir complejidad del modelo 🡪 Menos pesos/neuronas
* Tecnica de **regularización (L2, L1, Dropout)**

**Regularizacion L2**

🡪 Decaimiento de pesos

🡪 Nueva función de costo

🡪 Funciona mejor cuando la mayoría de atributos **son relevantes**

🡪 No se aplica al bias

Texto, Esquemático

Descripción generada automáticamente

**Regularizacion L1**

🡪 Ayuda a **seleccionar características importantes** y convertir el resto en ceros

🡪 Crea **vectores dispersos de pesos** como resultado

Imagen de la pantalla de un celular de un mensaje en letras blancas

Descripción generada automáticamente con confianza baja

**Dropout**

🡪 No modifica la función de costo 🡪 Modifica **la arquitectura de la de la red**

**Proceso:**

1. Selecciona **neuronas aleatorias que no participaran de la sig. Iteración** y las **borra** temporalmente
2. **Actualiza** los pesos
3. **Restaura las neuronas** borradas
4. **Repite** hasta estabilizar

**Velocidad de aprendizaje**

🡪 Determinada por alfa 🡪 Si es muy alto la red se puede desestabilizar

🡪 Se incorpora a la modificación de pesos **un termino que incluya la proporción del ultimo cambio** 🡪 **Momento**

**Clase 4b – Multiperceptron**

**Evaluación del modelo**

**Clasificación binaria**

**🡪** Reulstados se etiquetan como positivos o negativos

🡪 Se realiza la siguiente matriz de confusión:

Tabla

Descripción generada automáticamente

* **TVP**: Tasa de verdaderos positivos 🡪 VP / P
* **TFP**: Tasa de falsos positivos 🡪 FP / N
* **TVN**: Tasa de verdaderos negativos 🡪 VN / N