

Memoria

1.1.

Describe el algoritmo de retropropagación implementado.

El algoritmo de retropropagación es un método utilizado en las redes neuronales artificiales para optimizar los pesos de las conexiones en función del error obtenido en la salida. Este proceso se lleva a cabo en dos fases principales: propagación hacia adelante y retropropagación del error. Y en nuestra implementación, las clases son:

- Clase Conexion: Representa la conexión entre dos neuronas. Guarda el peso actual de la conexión, el peso anterior (posiblemente para momentum o verificación), y la neurona destino.
- Clase Neurona: Representa una neurona individual. Almacena su nombre, valor actual (activación), y una lista de conexiones a otras neuronas.
- Clase Capa: Agrupa múltiples neuronas y permite realizar operaciones como conectar neuronas entre capas, activar las neuronas en la capa (calcular sus salidas) y propagar valores entre neuronas.
- Clase Retropropagacion: Orquesta el proceso de entrenamiento de la red. Contiene métodos para propagar valores de entrada hacia adelante a través de la red, calcular errores y ajustar pesos utilizando retropropagación, e inicializar los valores de entrada de la red.

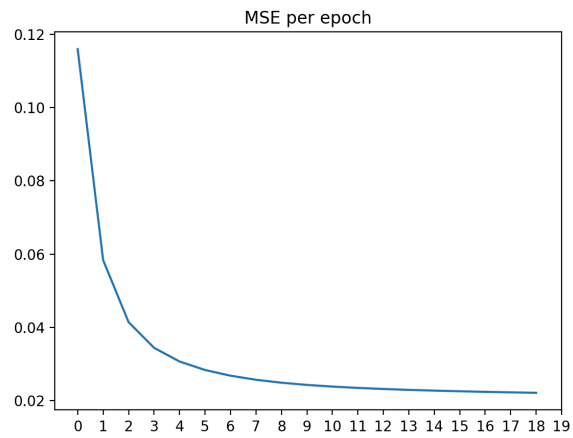
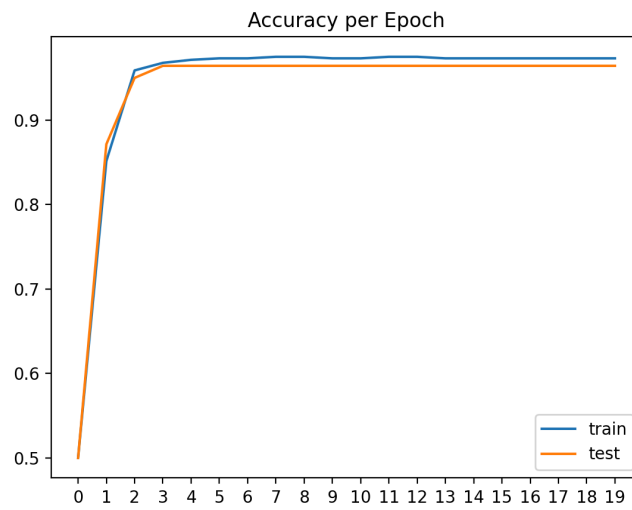
2.1

Describid todos los experimentos llevados a cabo y los resultados obtenidos.

Usamos 2 neuronas ocultas, 0.1 tasa de aprendizaje y nuestros resultados muestran el entrenamiento y las pruebas para cada problema de 1 a 20 épocas.

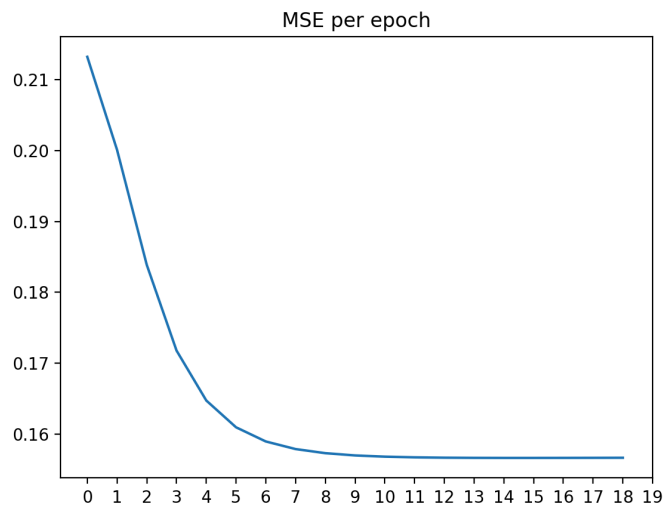
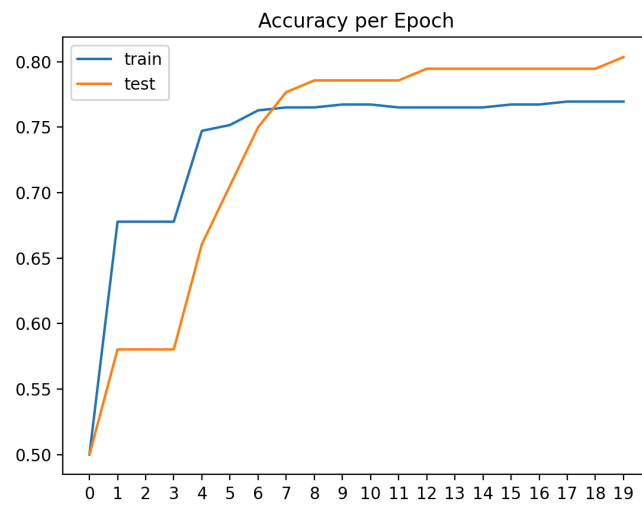
Problema real 1:

Se necesitaron muy pocas épocas para alcanzar un alto nivel de precisión, porque la naturaleza del problema era fácil de resolver con 2 neuronas ocultas.



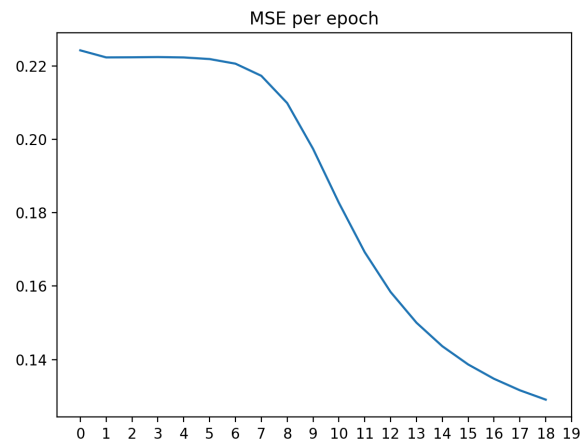
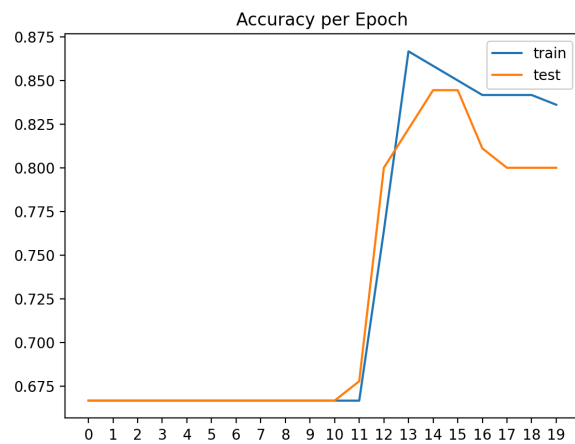
Problema real 2:

Fueron necesarias muchas épocas para alcanzar un alto nivel de precisión, ya que el problema es más difícil. También hay evidencia de sobreentrenamiento en los gráficos.



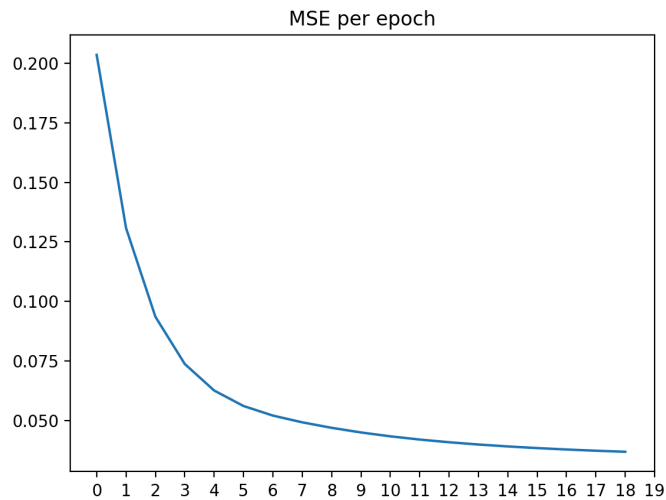
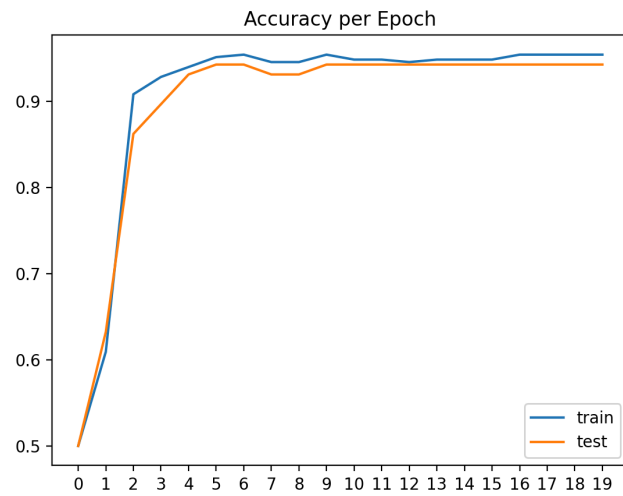
Problema real 3:

Se necesitaron muchas épocas para alcanzar un alto nivel de precisión, tal vez más neuronas ocultas producirían mejores resultados.



Problema real 5:

Se necesitaron muy pocas épocas para alcanzar un alto nivel de precisión, porque la naturaleza del problema era fácil de resolver con 2 neuronas ocultas.



3.1

Explicad la normalización aplicada a los datos, cómo se ha hecho y porqué se hace.

Proceso de Normalización Aplicado:

- Se calculó la media y la desviación estándar para cada atributo del conjunto de datos de entrenamiento. Estas estadísticas se utilizan para ajustar la escala de los datos.
- La transformación aplicada a los datos fue la estandarización, donde cada atributo en los datos de entrenamiento se reescala de la siguiente manera:

$$\text{Valor Normalizado} = (\text{Valor Original} - \text{Media}) / \text{Desviación Estandar}$$

- Esta misma transformación se aplica a los datos de test, utilizando la media y la desviación estándar calculadas del conjunto de entrenamiento.

Razón para Normalizar:

- La normalización es crucial porque los atributos pueden tener rangos muy diferentes y distintas unidades de medida. Sin normalización, los atributos con rangos más amplios podrían dominar el proceso de aprendizaje, causando que el modelo sea ineficiente o sesgado.
- Además, algoritmos como el descenso del gradiente (utilizado en la retropropagación dentro de las redes neuronales) convergen más rápidamente cuando todos los atributos están en escalas similares.