

Proyecto 1: Redes Neuronales Artificiales

Borge Chacon Brainer Antonio, Medinilla Robles Pedro, Obando Pereira David

I. 7.1 FUNCIONES DE EVALUACIÓN DEL DESEMPEÑO

I-A. Matriz de Confusión

Es una forma para visualizar el desempeño de un algoritmo de aprendizaje. Con las columnas de esta matriz representando el número de predicciones de cada clase y las filas como las instancias en la clase real. Permite entonces, ver los aciertos y errores que está teniendo el modelo en el proceso de aprendizaje. Compara el modelo real con el modelo predictivo. Diferencia el desempeño entre una categoría u otra. El número de predicciones correctas e incorrectas se resume con los valores de conteo y se desglosan por cada clase.

Términos:

- Positivo (P)
- Negativo (N)
- Verdadero Positivo (TP)
- verdadero Negativo (TN)
- Falso Positivo (FP)
- Falso Negativo (FN)

I-B. Métricas de Evaluación de Clasificación para Matriz de Confusión

- **Exactitud:** Es igual a la proporción de predicciones que el modelo clasificó correctamente.

$$Exact. = \frac{Num.Pred.Correctas}{Num.Tot.dePred.} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

- **Exhaustividad** (ó Sensibilidad): Proporción de la cantidad de instancias que se recuperaron realmente. Determina la proporción de positivos reales que se identificaron realmente.

$$Sensibilidad = \frac{TP}{TP + FN} \quad (2)$$

- **Especificidad (Tasa Negativa Real(TNR)):** Muestra la proporción de negativos reales que se identifican correctamente correctamente como tales. Es lo opuesto a la Sensibilidad.

$$Specificity = \frac{TN}{TN + FP} \quad (3)$$

- **Precisión (Valor Predictivo Positivo):** Mide la capacidad del modelo para clasificar correctamente los casos positivos. Se calcula dividiendo el número de verdaderos positivos entre la suma de los verdaderos positivos y los falsos positivos:

$$Precision = \frac{TP}{TP + FP} \quad (4)$$

- **Valor F1:** Medida que combina tanto la precisión como la exhaustividad en una sola métrica. Se calcula como la media armónica de la precisión y la exhaustividad.

$$F1 = \frac{2 \cdot Precision \cdot Sensibilidad}{Precision + Sensibilidad} \quad (5)$$

Algo importante en el código es que el componente (i,j) de la matriz de confusión representa el número de instancias que pertenecen a la clase i y que han sido clasificadas por el modelo como pertenecientes a la clase j. En otras palabras, cada fila de la matriz de confusión representa las instancias reales de una clase en particular, mientras que cada columna representa las instancias predichas por el modelo para esa misma clase.

II. 7.2 VISUALIZACIÓN DE LOS RESULTADOS BIDIMENSIONALES

Para visualizar los datos, se hizo el script softmaxvisualizer.m.

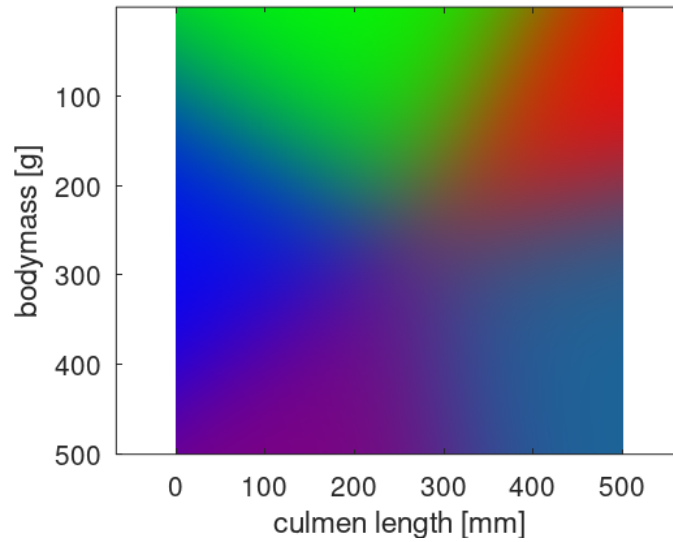


Figura 1: Salida ponderada para datos artificiales

III. 7.5 PROCEDIMIENTO

III-A. Diseño de experimentos con datos artificiales

- Evaluación de capas de activación Para este experimento se utilizó un learning rate de 0.01, beta2 de 0.99, beta1 de 0.9 y un mini batch de 32, dando como resultado la Figura 2.
En el script trainp1.m se encuentra esta prueba.

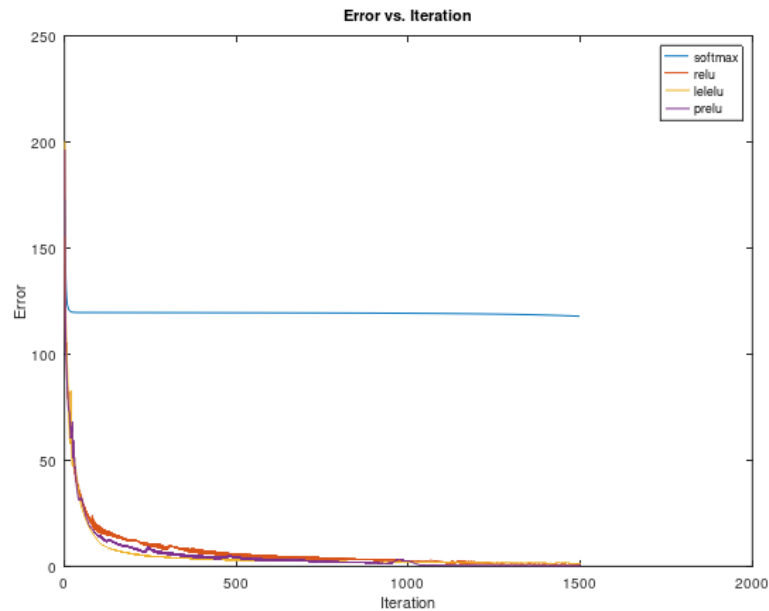


Figura 2: Pérdida en función de la iteración para las diferentes capas de activación

- Evaluación de funciones de pérdida

En el script trainp2.m se encuentra esta prueba.

Debido al tipo de datos, MAE y STD no representan bien el error, por lo cual en la Figura 3 no se ve bien, en la Figura 4 se ven mejor:

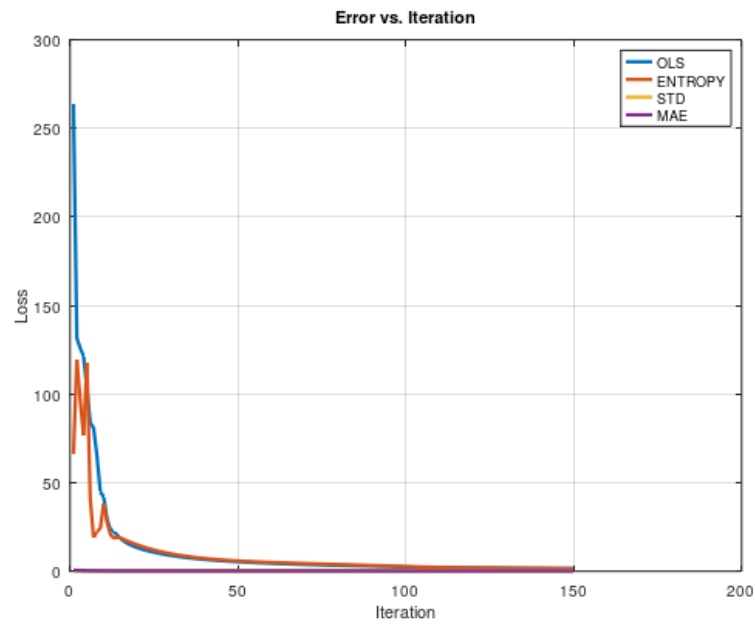


Figura 3: Pérdida en función de la iteración para los diferentes cálculos de error

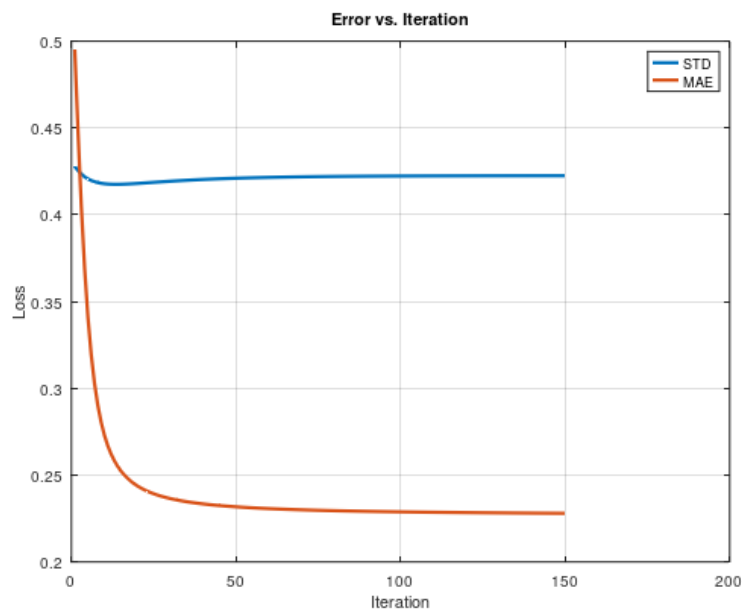


Figura 4: Pérdida en función de la iteración para MAE y STD

■ Evaluación de métodos de optimización

En el script trainp3.m se encuentra esta prueba.

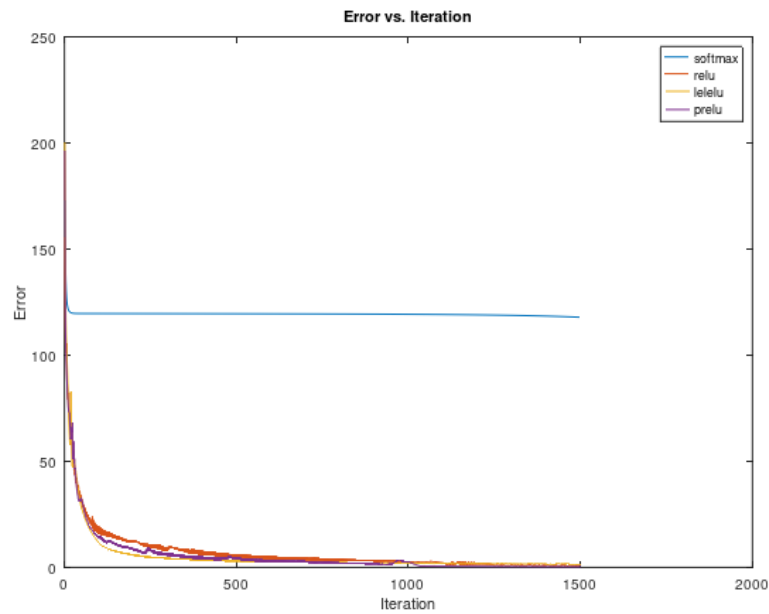


Figura 5: Pérdida en función de la iteración para los diferentes métodos de optimización

■ Evaluación de hiperparámetros

Para la variación de hiperparámetros se inicio con un alpha de 0.5, beta2 de 0.99, beta1 de 0.9 y minibatch de 32, estos se fueron reduciendo 10 % por cada época y minibatch se fue reduciendo de 5 en 5.

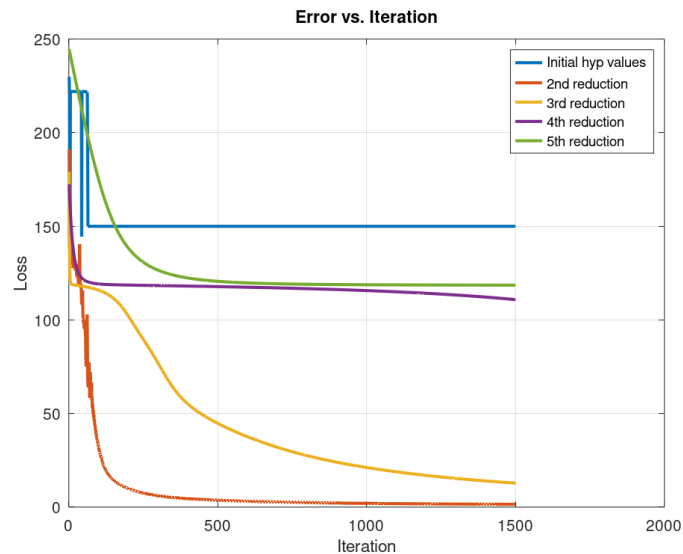


Figura 6: Pérdida en función del cambio de los hiperparámetros

Como se puede notar, en la segunda reducción hubo menor error, aquí el alpha es de 0.05, beta2 de 0.099, beta1 de 0.09 y minibatch de 27. En el script trainp4.m se encuentra esta prueba.

III-B. Diseño de experimentos con datos de pingüinos

■ Evaluación de arquitecturas de red neuronal

Para este experimento se utilizó relu, prelu y lelelu con batch, dando como resultado la Figura 7. En el script trainpen.m se encuentra esta prueba.

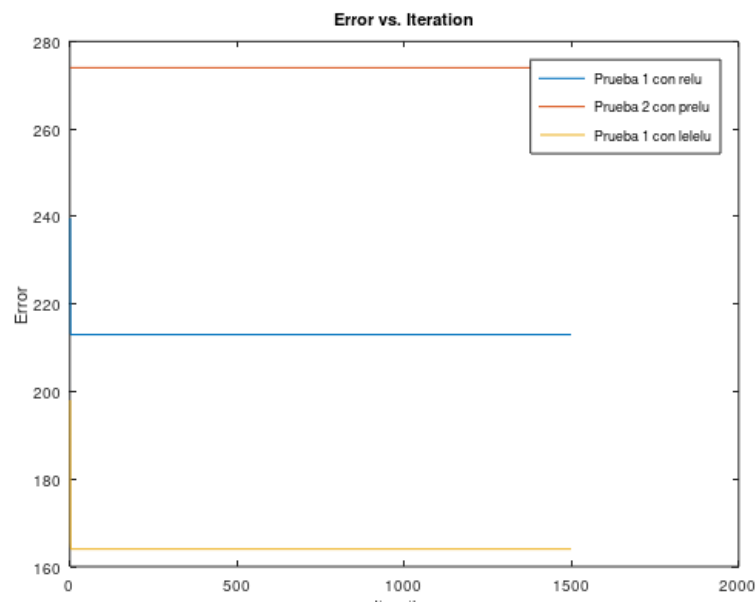


Figura 7: Pérdida en función de la iteración para las diferentes arquitecturas

■ Matriz de confusión para las 3 clases

Resultados de clasificacion de la matriz de confusion			
Clase	Precision	Exhaustividad	Exactitud
1	0.341463	0.500000	0.264706
2	0.000000	0.000000	0.264706
3	0.200000	0.173913	0.264706

Figura 8: Criterios de la matriz de confusión para la arquitectura con lelelu

- Clase ganadora y ponderada

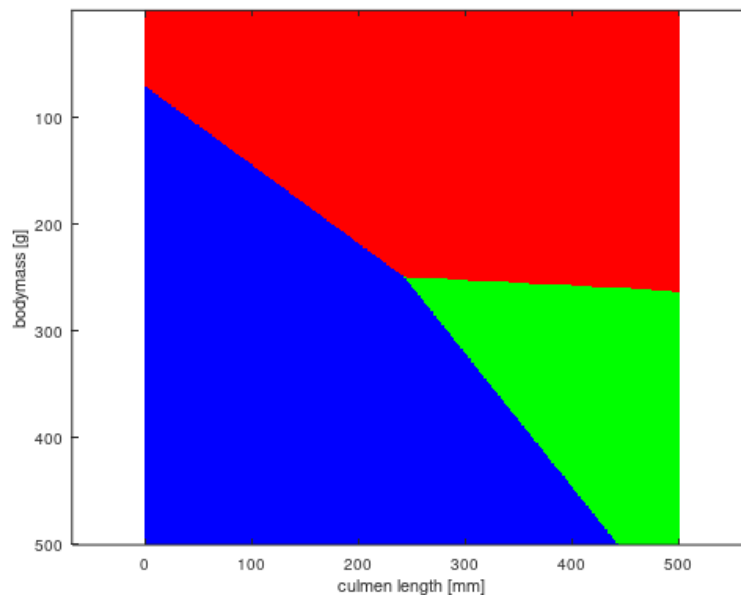


Figura 9: Clase ganadora

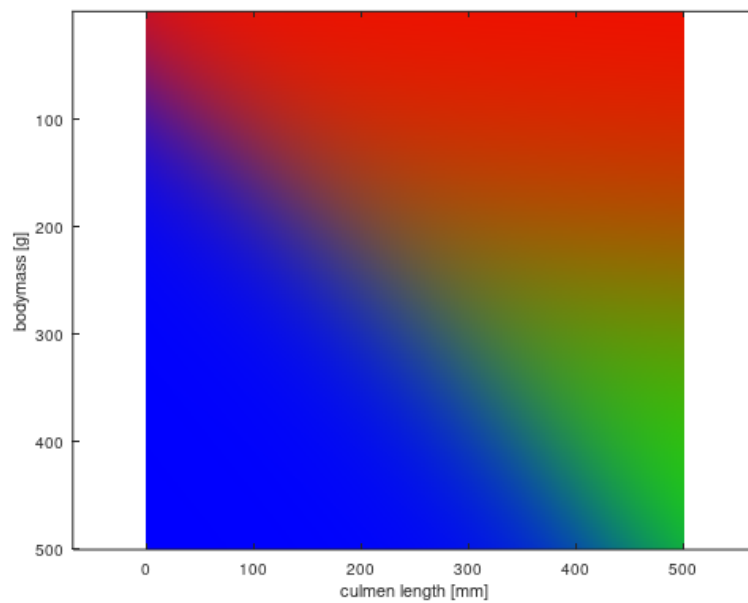


Figura 10: Clase ponderada