

Reporte

Reporte de Funciones, Modelos Personalizados y Ecuaciones diferenciales

Carolina Arce Cruz

Facultad de Ciencias Físico Matemáticas
Benemérita Universidad Autónoma de Puebla

23 de marzo de 2024

Resumen

En este reporte se explica la implementación de Keras para el diseño de una red para transformar imágenes a color, reproducir funciones y sus respectivas gráficas y entrenar una red para dar solución a una ecuación diferencial.

1. Objetivo

1. Diseño de una capa en keras que transforme imágenes a color en escala de grises.
2. Entrenar una red que reproduzca funciones y graficarlas.
3. Diseño de una capa entrenable que represente un polinomio grado 3.
4. Entrenar una red neuronal que de solución de dos ecuaciones diferenciales.

2. Resultados y análisis

Ejercicio 1

En este primer ejercicio hice dedos maneras el código y pienso que salieron bien, pero el primer código creo que realmente no cambió nada,

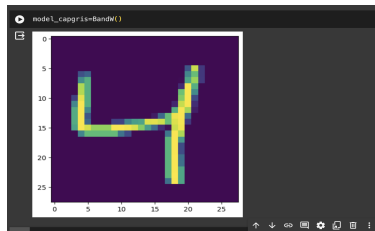


Figura 1. Resultado del primer intento

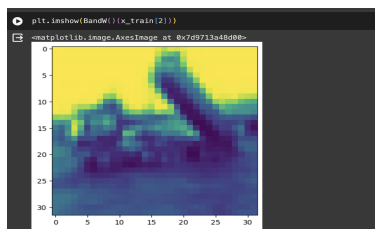


Figura 2. Resultado del primer intento

```

def compile(optimizer="adam", metrics=["loss"]):
    model.compile(optimizer=optimizer, metrics=metrics)
    history = model.fit(x_train, y_train, epochs=1000, verbose=0)
    plt.plot(history.history["loss"])

def train():
    # Load the data
    x_train = x_train.astype('float32')
    y_train = y_train.astype('float32')
    x_test = x_test.astype('float32')
    y_test = y_test.astype('float32')

    # Compile the model
    compile(optimizer="adam", metrics=["loss"])

    # Train the model
    history = model.fit(x_train, y_train, epochs=1000, verbose=0)

    # Evaluate the model
    loss = model.evaluate(x_test, y_test, verbose=0)
    print("Test loss: %f" % loss)

if __name__ == '__main__':
    train()

```

Figura 5. Código del inciso b)

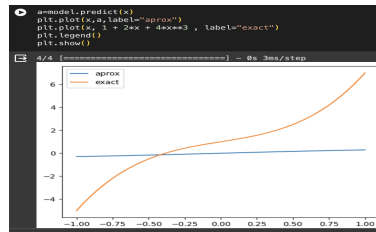


Figura 6. Gráfica del inciso b)

Quiero hacer notar que en la primera imagen no se ve el error porque lo volví a correr pero se quedó trabado y en la tercera imagen aparece el error. Por este error en ambas gráficas no se visualiza bien.

Ejercicio 3

Este ejercicio pienso que fue de los mejores que salió en general, el código ya está en github, solo adjunto la gráfica

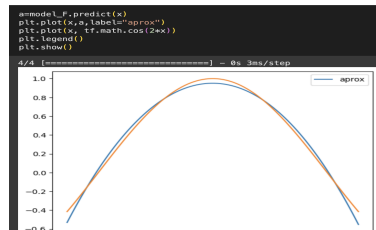


Figura 7. Gráfica de la función

Ejercicio 4

Este ejercicio también me costó trabajo y en las gráficas se muestra. No me salía error, pero no fueron los mejores resultados. También quiero agregar que en la gráfica 4.2 la aproximación no se ve bien porque cuando volví a correr el código, tardó mucho y se quedó trabado.

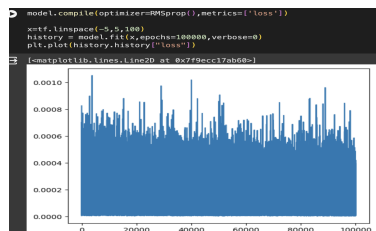


Figura 8. Gráfica 4.1

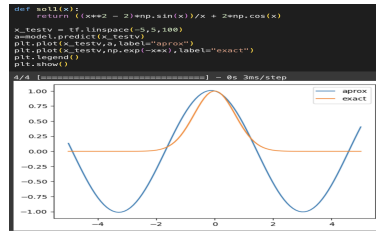


Figura 9. Gráfica 4.2

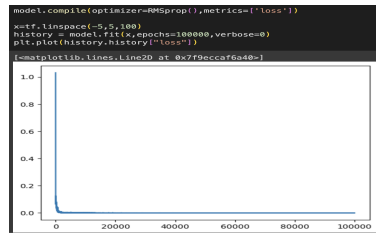


Figura 10. Gráfica 4.1

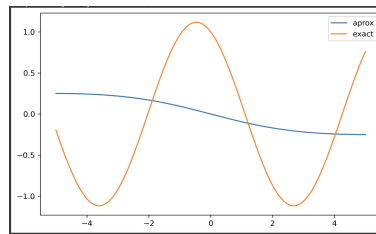


Figura 11. Gráfica 4.2

3. Conclusiones

Pienso que pude hacer más pruebas e intentar diferentes valores para las capas, además de tiempo en general porque me tardaba mucho en algunos códigos y se me quedaba trabado cuando corría los códigos. Pero en general pienso que no salió "mal", pero se puede mejorar.