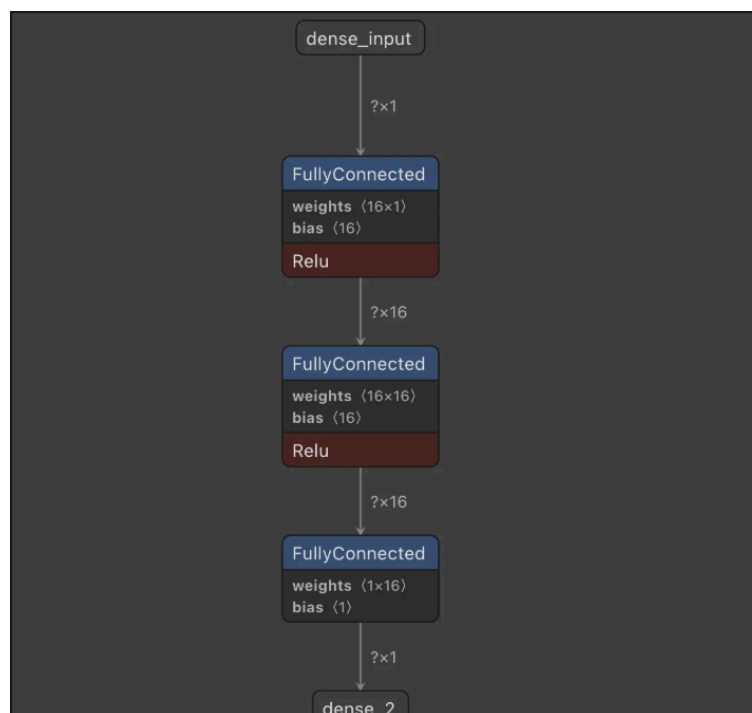


Lab 1: Ejercicio 2 Sistemas Embebidos

Alumnos: Diego Giordano - Benjamín Rogers

Profesor: Jorge Gomez

Implementación propia de algoritmo de hello world



Topología modelo ejemplo hello world

Se copió el funcionamiento del modelo de ejemplo en TF micro mediante multiplicaciones vectoriales directamente en C, específicamente se toma un input de 1 dimensión, se multiplica por un vector de 16 pesos, luego estos nuevos 16 valores por una matriz de 16x16 y luego estos 16 valores a una sola neurona. Se copiaron los pesos identicos obtenidos al revisar el modelo ejemplo en Netron. El algoritmo puede apreciarse mejor en el código.

A continuación se presentan las diferencias de tiempos de inferencia de ambos estilos:

```

the binary image header.
I (283) main_task: Started on CPU0
I (293) main_task: Calling app_main()
Neural Network Inference Test
Architecture: 1 -> 16 -> 16 -> 1
=====

Running inference tests:
Input          Output          Time (µs)
-----
0.00           0.026405        82
0.50           0.453988        62
1.00           0.863044        62
-0.50          -0.509319       62
-1.00          -1.048297       62
2.00           0.887233        62
-2.00          -2.208764       62
0.75           0.672542        62
-0.25          -0.241457       62
1.50           0.981648        62

Performance benchmark (1000 iterations):
Total time for 1000 inferences: 53585 µs
Average time per inference: 53.58 µs
Inferences per second: 18662

Neural network inference complete!
I (373) main_task: Returned from app_main()

Done

```

Proyecciones y tiempo de inferencia del modelo propio

```
(base) diegogiordano@Diegos-MacBook-Air-2 hello_world % source ~/.zprofile && source ~/.
&& idf.py flash -p /dev/cu.usbserial-110 monitor
x_value: 1.570796, y_value: 1.042060, inference_time: 71 us
x_value: 1.884956, y_value: 0.957340, inference_time: 70 us
x_value: 2.199115, y_value: 0.821787, inference_time: 71 us
x_value: 2.513274, y_value: 0.533738, inference_time: 70 us
x_value: 2.827433, y_value: 0.237217, inference_time: 71 us
Inferences per second: 2.22
x_value: 3.141593, y_value: 0.008472, inference_time: 72 us
x_value: 3.455752, y_value: -0.304993, inference_time: 71 us
x_value: 3.769912, y_value: -0.533738, inference_time: 70 us
x_value: 4.084070, y_value: -0.779427, inference_time: 71 us
x_value: 4.398230, y_value: -0.965812, inference_time: 70 us
x_value: 4.712389, y_value: -1.109837, inference_time: 71 us
x_value: 5.026548, y_value: -0.982756, inference_time: 70 us
x_value: 5.340708, y_value: -0.745539, inference_time: 71 us
x_value: 5.654867, y_value: -0.533738, inference_time: 71 us
x_value: 5.969026, y_value: -0.355825, inference_time: 71 us
Inferences per second: 2.00
x_value: 0.000000, y_value: 0.000000, inference_time: 72 us
x_value: 0.314159, y_value: 0.372770, inference_time: 70 us
x_value: 0.628319, y_value: 0.559154, inference_time: 69 us
x_value: 0.942478, y_value: 0.838731, inference_time: 70 us
x_value: 1.256637, y_value: 0.965812, inference_time: 69 us
x_value: 1.570796, y_value: 1.042060, inference_time: 71 us
x_value: 1.884956, y_value: 0.957340, inference_time: 70 us
x_value: 2.199115, y_value: 0.821787, inference_time: 71 us
x_value: 2.513274, y_value: 0.533738, inference_time: 70 us
x_value: 2.827433, y_value: 0.237217, inference_time: 71 us
Inferences per second: 2.00
x_value: 3.141593, y_value: 0.008472, inference_time: 72 us
x_value: 3.455752, y_value: -0.304993, inference_time: 71 us
x_value: 3.769912, y_value: -0.533738, inference_time: 70 us
x_value: 4.084070, y_value: -0.779427, inference_time: 71 us
x_value: 4.398230, y_value: -0.965812, inference_time: 70 us
x_value: 4.712389, y_value: -1.109837, inference_time: 71 us
x_value: 5.026548, y_value: -0.982756, inference_time: 70 us
x_value: 5.340708, y_value: -0.745539, inference_time: 71 us
x_value: 5.654867, y_value: -0.533738, inference_time: 71 us
x_value: 5.969026, y_value: -0.355825, inference_time: 71 us
Inferences per second: 2.00
```

Proyecciones y tiempos de inferencia del ejemplo de TF micro

Primero que nada destacar que el código del ejemplo debió ser modificado para imprimir el tiempo de inferencia.

Al aproximar el tiempo de inferencia promedio del ejemplo se obtienen 70.5 microsegundos mientras que el algoritmo propio promedió 62 microsegundos (a excepción de la primera inferencia que se puede atribuir a un tiempo de iniciación u overhead), por lo tanto se puede declarar más rápido el algoritmo propio, aunque la diferencia de tiempo es imperceptible al ojo humano, para contextos críticos puede ser un elemento a tomar en consideración.