



**DEPARTAMENTO  
DE COMPUTACION**

Facultad de Ciencias Exactas y Naturales - UBA

# Trabajo Final

30 de julio de 2017

Organización del Computador II

Integrante	LU	Correo electrónico
Costa, Manuel	COMPLETAR	COMPLETAR
Gatti, Mathias	477/14	mathigatti@gmail.com



**Facultad de Ciencias Exactas y Naturales**  
**Universidad de Buenos Aires**

Ciudad Universitaria - (Pabellón I/Planta Baja)

Intendente Güiraldes 2160 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Fax: (54 11) 4576-3359

<http://www.fcen.uba.ar>

# Índice

<b>0. Introducción</b>	<b>3</b>
0.1. Manual de usuario . . . . .	3
<b>1. Tópicos Teóricos</b>	<b>4</b>
<b>2. Resultados</b>	<b>5</b>
<b>3. Conclusiones</b>	<b>6</b>

## 0. Introducción

La idea es implementar una red neuronal sigmoidea. Debido a la intensa cantidad de computos que podrían realizarse de forma paralela (Principalmente operaciones matriciales) pensamos como foco principal del proyecto estudiar que tanto se podría mejorar la performance temporal del programa aprovechando SIMD y demás beneficios que podamos sacar a partir de la utilización de ASSEMBLER.

Decidimos que la red resuelva la clásica tarea de reconocimiento de digitos manuscritos, utilizando el dataset MNIST. Escogimos esta tarea por ser un problema de tamaño razonable para tratar con el hardware del que disponemos, pero que a su vez es lo suficientemente grande como para permitir un análisis interesante. La idea de nuestro trabajo es implementar una red neuronal sigmoidea. Debido a la intensa cantidad de computos que podrían realizarse de forma paralela (Principalmente operaciones matriciales) pensamos como foco principal del proyecto estudiar que tanto se podría mejorar la performance temporal del programa aprovechando SIMD y demás beneficios que podamos sacar a partir de la utilización de ASSEMBLER. Para darle una utilidad a la red y poder testear su eficacia decidimos aplicarla a un problema clásico, reconocer caracteres numéricos, entrenaremos al programa con un gran set de datos y luego corroboraremos su buen funcionamiento reconociendo digitos nuevos.

### 0.1. Manual de usuario

–COMPLETAR MANUAL DE USO DEL PROGRAMA–

- \* herramientas especiales de desarrollo utilizadas si las hubiese

Cosas que me parece que podriamos poner brevemente

- \* uso de makefile

- \* explicacion de tests

# 1. Tópicos Teóricos

–COMPLETAR CON TOPICOS TEORICOS–

## 2. Resultados

Hacer un cuadro de resultados comparando la implementación contra la mejor compilación con gcc -O3, para diferentes casos de algoritmos

Task	Waiting-time				Latencia				Turnaround			
	RR	FIFO	SJF	RSJF	RR	FIFO	SJF	RSJF	RR	FIFO	SJF	RSJF
0	39	1	1	20	1	1	1	1	44	11	11	30
1	37	12	30	35	6	12	30	31	57	32	50	55
2	28	29	8	2	7	29	8	2	33	34	13	7
3	30	35	14	8	12	35	14	8	35	40	19	13
4	32	46	20	14	17	46	20	14	37	46	25	19

Tabla 1 Cuadro Choreado 1

Scheduler	Waiting-time	Latencia	Turnaround
RR	33.5	8.6	41.2
FIFO	24.6	24.6	32.6
SJF	14.6	14.6	23.6
RSJF	15.8	11.2	24.8

Tabla 2 Cuadro Choreado 2

### **3. Conclusiones**

–COMPLETAR CONCLUSION–