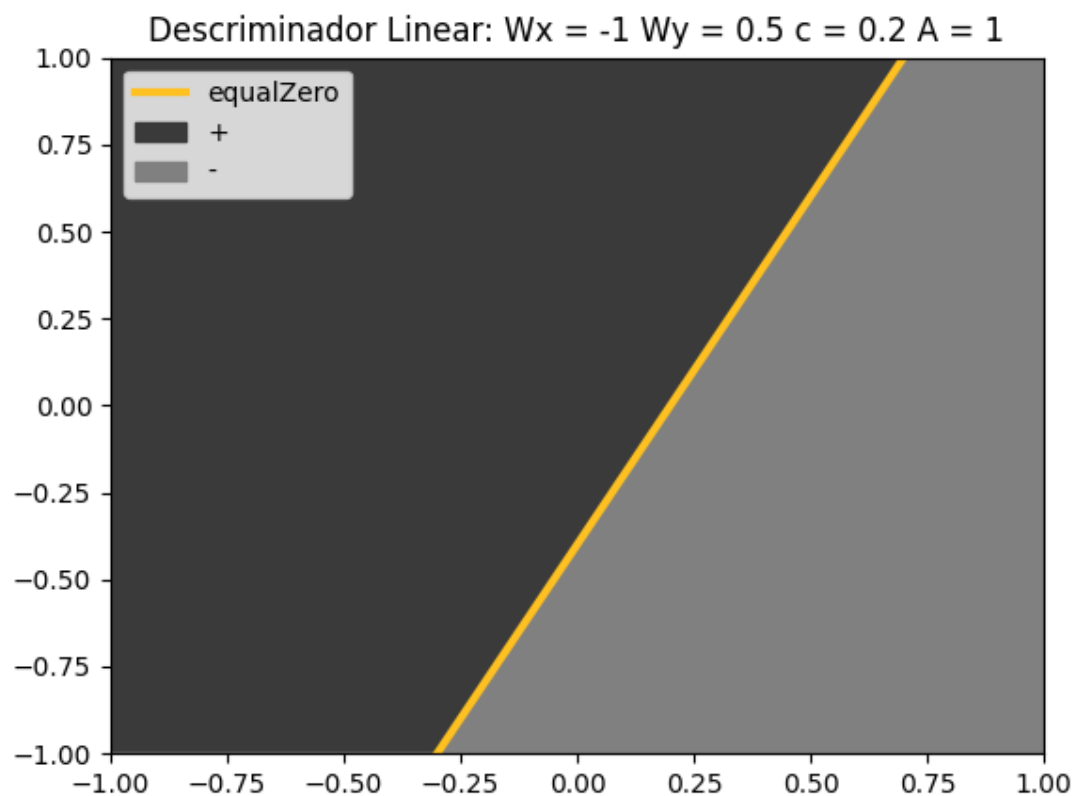


P6 - Resultados

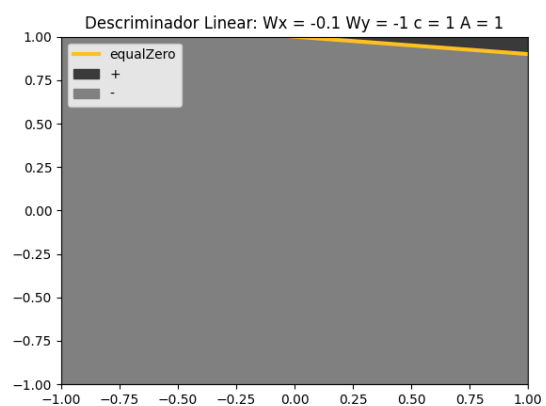
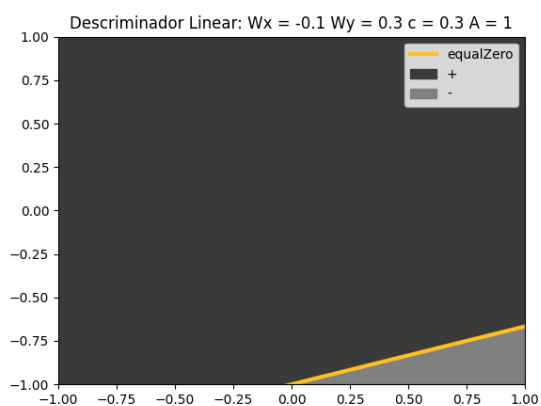
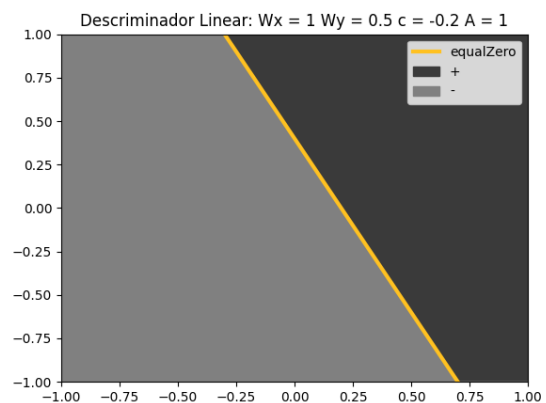
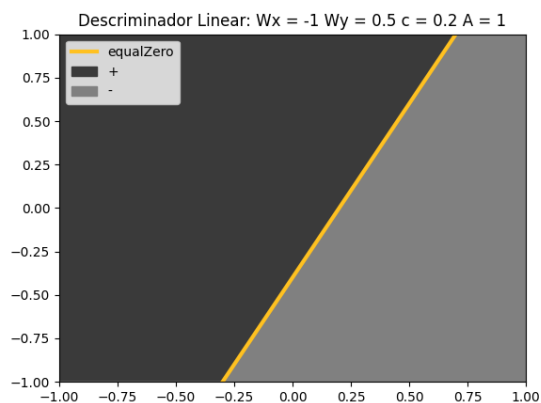
Parte A

Na etapa desse projeto, foi requisitado a obtenção de uma imagem similar à figura 3 do CDT-25.

Dessa forma, fomos capazes de implementar tal imagem a seguir:



Variando os valores de W_x , W_y e c , temos outros possíveis discriminadores.

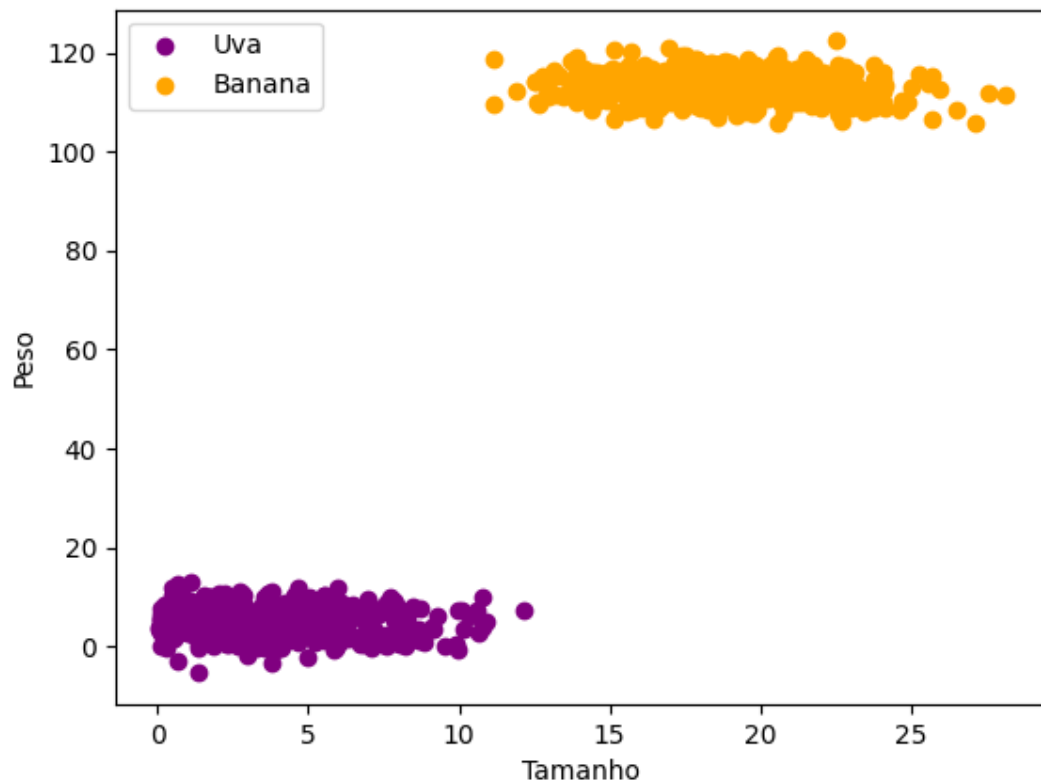


Ainda nos foi pedido um discriminador que fosse capaz de separar entre uvas e bananas com base em seus pesos e tamanhos. Dessa forma, construímos uma base de dados com 500 amostras partindo de uma distribuição normal em torno dos pesos e tamanhos típicos dessas frutas conforme a tabela abaixo.

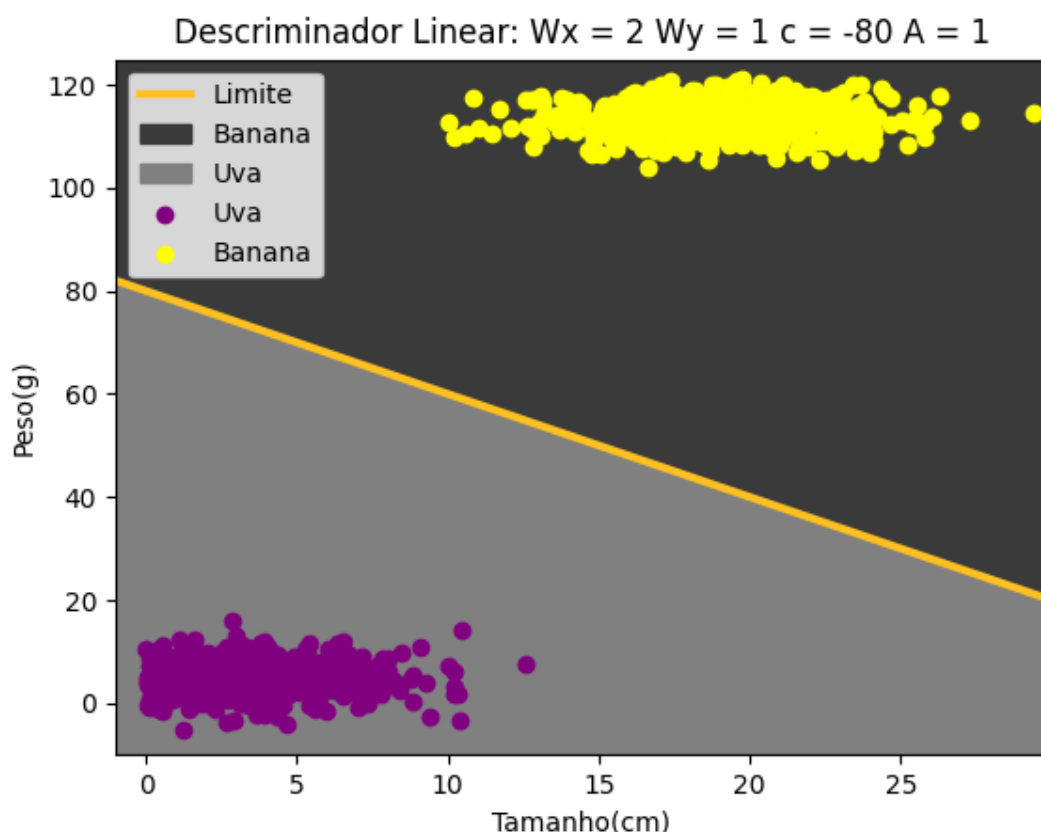
Frutas

Frutas	Uva	Banana
Tamanho [cm]	3	19
Peso [g]	5.1	113.39

Tal distribuição nos retorna o seguinte conjunto de dados:



Partindo disso, implementamos um discriminador linear que separa o primeiro quadrante do plano cartesiano em duas regiões referêntes as respectivas frutas, conforme o mostrado na figura a seguir.



Parte B

Nessa seção, implementamos um reconhecedor de dígitos de 0 a 9 baseado no banco de dados MNIST. O primeiro passo foi a obtenção das imagens, seguido pela separação do conjunto de imagens em dois subconjuntos para o treinamento e teste. Os vetores de pesos de cada um dos dígitos foram construídos, em primeira aproximação, pela escolha de uma das imagens do subconjunto de treinamento. Escolhendo uma imagem como padrão do dígito abordado, empilhamos suas colunas transformando-a em um vetor 1D. Tal vetor é tido como a referência de pesos desse dígito. Repetimos esse processo para os demais dígitos, possibilitando a inferência das imagens do subconjunto de testes. Nessa etapa, a partir da escolha de uma imagem teste como estímulo de nossa rede, calculamos a "proximidade" dos dois vetores como o produto escalar (que projeta um vetor no outro) conforme a equação abaixo.

$$s = \langle \vec{x}, \vec{w} \rangle$$

A partir do valor do produto escalar s de nossa imagem com todos os vetores de peso, escolhemos o maior valor (que implica em um maior alinhamento dos vetores) como o dígito inferido.

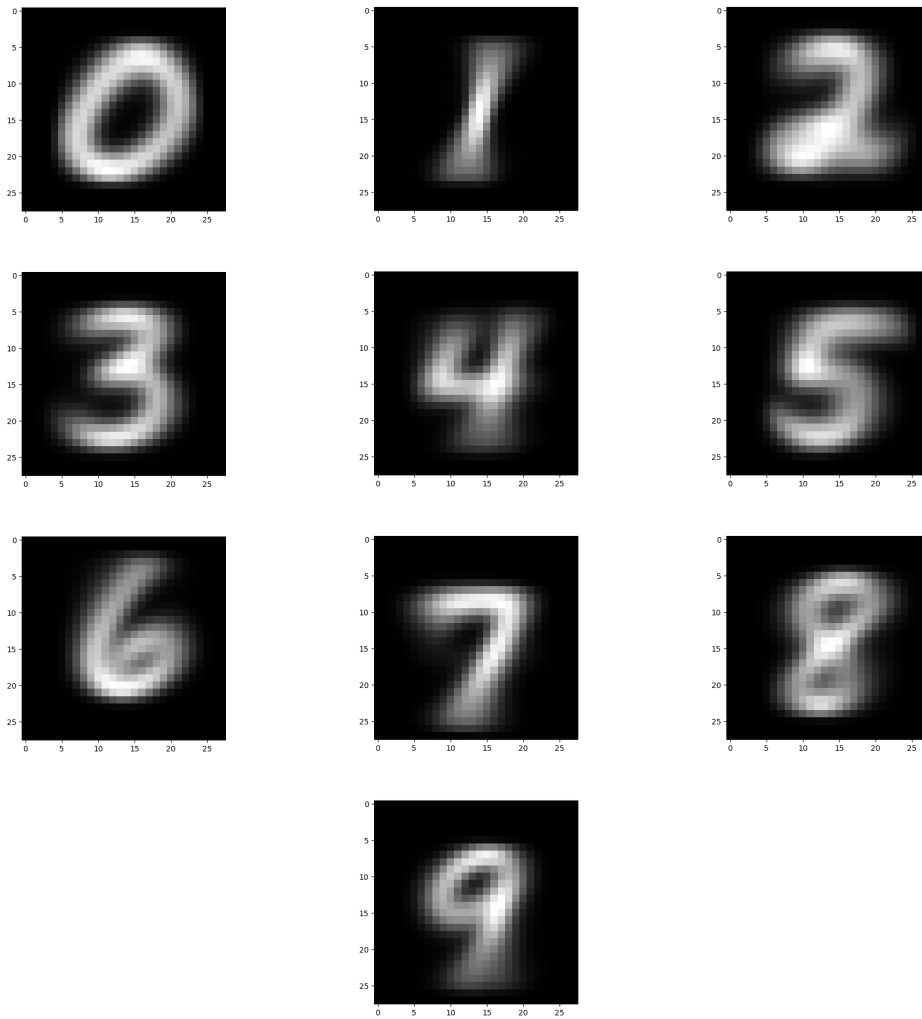
Tendo os rótulos corretos de cada uma das imagens, pudemos construir a matriz de confusão demonstrada a seguir.

		One Weighs									
		0	1	2	3	4	5	6	7	8	9
Inferido	0	323	326	315	317	278	264	288	282	311	314
	1	296	333	296	311	287	258	315	297	301	312
	2	310	316	277	290	275	295	312	348	293	322
	3	269	350	324	309	308	271	294	319	301	299
	4	316	357	286	282	296	252	269	317	295	262
	5	303	337	346	328	298	248	271	311	312	287
	6	295	332	304	318	294	285	322	347	291	327
	7	277	328	289	299	286	271	320	307	289	265
	8	284	312	287	304	296	287	271	315	283	268
	9	289	328	286	300	298	281	281	315	299	290
		Real									

Matriz confusão considerando pesos com somente um dígito.

Vemos que essa rede tem pouca capacidade de inferência pois um cada dígito é determinado de forma aproximadamente igual com todos os outros.

Com o intuito de melhorar nossa rede, passamos a definir o vetor peso como a média de todas as matrizes de um certo dígito do conjunto de treinamento, como mostra a imagem abaixo.



Nela é possível ver que ocorre um processo de perda de foco nos dígitos, uma vez que temos a contribuição de varios outros digitos com pequenas variações de formato e posição. Isso nos garante uma maior generalização do vetor de pesos que adotamos como referência, possibilitando então um processo de inferência mais eficiente.

Dessa forma, seguindo o mesmo processo adotado anteriormente, chegamos à seguinte matriz de confusão:

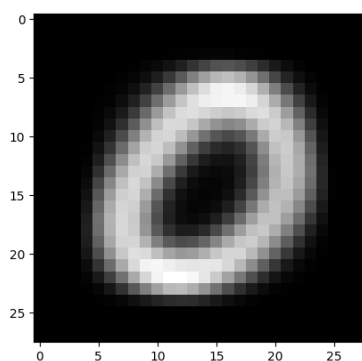
		noErodeZero									
		0	1	2	3	4	5	6	7	8	9
Inferido	0 - 2894	5	436	376	332	1443	459	418	185	321	
	1	0	1586	1	0	0	0	0	1	0	1
	2	4	13	2022	36	21	32	42	26	16	25
	3	1	21	83	2152	15	325	4	30	94	66
	4	0	0	7	0	1432	4	0	17	0	26
	5	0	0	0	0	0	0	0	0	0	0
	6	9	10	111	17	115	29	2291	6	18	5
	7	0	1	8	2	2	0	0	1896	0	17
	8	53	1680	336	465	590	856	147	562	2653	702
	9	1	3	6	10	409	23	0	202	9	1783
		Real									

Nela vemos que a chance de termos uma inferência correta tende a ser mais recorrente. Contudo, ainda vemos que o dígito 0 aparece com maior frequência em todos os dígitos. Analisando de forma mais detalhada, nos deparamos os seguintes valores das normas dos vetores de peso.

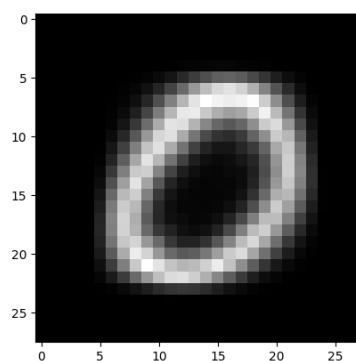
Norma dos vetores de peso

Aa Dígitos	≡ 0	≡ 1	≡ 2	≡ 3	≡ 4	≡ 5	≡ 6	≡ 7	≡ 8	≡ 9
<u>Norma</u>	2138	1385	1802	1816	1623	1578	1806	1620	1901	1686

Fica claro que a norma do vetor correspondente ao padrão de pesos do dígito 0 é maior do que a dos outros dígitos. Dessa forma, é natural que o produto escalar entre os vetores de estímulo e este seja maior o que implica em uma inferência de um maior número de valores 0. Essa maior norma nos informa que a matriz do 0 contem mais entradas não nulas, ou seja, o número 0 apresentado na matriz das médias de todos os dígitos 0 tem contorno mais espesso do que as os demais dígitos.



Matriz do dígito 0 da média de todos os sinais de teste



Matriz do dígito 0 após erosão.

Contornamos esse problema a partir da aplicação do processo de erosão da matriz média. O resultado pode ser observado acima (imagem da direita).

Os valores das novas normas estão apresentados na matriz abaixo.

Norma dos vetores de peso (zero erodido)

<u>Aa</u> Name	0	1	2	3	4	5	6	7	8	9
<u>Norma</u>	1507	1385	1802	1816	1623	1578	1806	1620	1901	1686

Vemos que todos eles apresentam normas na mesma ordem de grandeza. Dessa forma seguimos para inferência dos dígitos de teste com as novas matrizes de peso, obtendo a seguinte matriz de confusão:

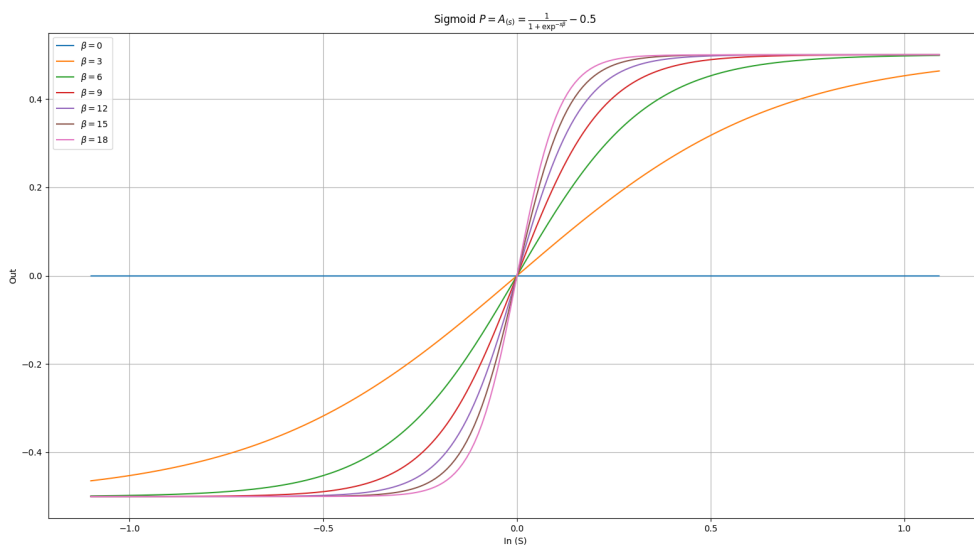
		ErodeZero									
		0	1	2	3	4	5	6	7	8	9
Inferido	0	1689	0	1	0	0	2	1	26	0	4
	1	0	1586	1	0	0	0	0	1	0	1
	2	176	13	2353	75	53	50	105	29	19	79
	3	176	21	119	2423	15	856	7	55	137	75
	4	1	0	9	0	1612	6	0	38	1	46
	5	28	0	0	0	0	74	1	0	0	0
	6	306	11	133	34	193	96	2650	8	25	19
	7	0	1	8	2	2	2	0	2153	0	21
	8	585	1684	379	514	610	1569	179	588	2784	766
	9	1	3	7	10	431	57	0	260	9	1935
		Real									

Nela é possível perceber que os falsos zeros tem menor frequência e portanto aumentamos o número de acertos dos demais dígitos.

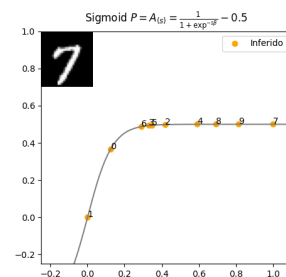
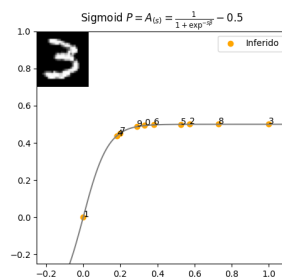
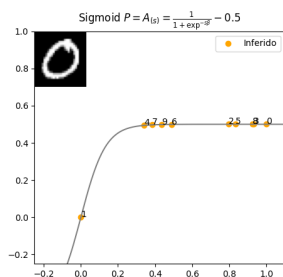
Seguindo para a parte final, implementamos a ativação sigmóide, cuja equação pode ser observada na sequência.

$$A_{(s)} = \frac{1}{1 + e^{-\beta s}} - 0.5$$

O gráfico de tal função com diferentes valores de β é dado por:



A ativação por sigmóide leva em conta o valor da função $A(s)$ para a categorização com base no valor de cada produto escalar entre os vetores de peso e estímulo. Dessa forma, fomos capazes de observar o seguinte comportamento da função sigmóide a partir dos dados de projeção de nossos vetores:



Nessas figuras, podemos observar o nível de ativação de nosso neurônio a partir da matriz de entrada como estímulo. Vemos que nesses casos, a ativação ocorreu de forma satisfatória para cada número.

Nosso processo resultou na seguinte matriz de confusão:

		sigmoidTrigger									
		0	1	2	3	4	5	6	7	8	9
Inferido	0	1689	0	1	0	0	2	1	26	0	4
	1	0	1586	1	0	0	0	0	1	0	1
	2	176	13	2353	75	53	50	105	29	19	79
	3	176	21	119	2423	15	856	7	55	137	75
	4	1	0	9	0	1612	6	0	38	1	46
	5	28	0	0	0	0	74	1	0	0	0
	6	306	11	133	34	193	96	2650	8	25	19
	7	0	1	8	2	2	2	0	2153	0	21
	8	585	1684	379	514	610	1569	179	588	2784	766
	9	1	3	7	10	431	57	0	260	9	1935
		Real									

Tal matriz apresenta o mesmo resultado da ativação pelo maior valor de projeção, o que é esperado pois escolhemos o maior valor de $A_{(s)}$ dentre os calculados como a inferência correta. Uma vez que a função é crescente, maiores valores de s implicam em maiores valores de $A_{(s)}$.