Redes Neuronales

Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Trabajo Práctico Número 1

BackPropagation

Integrante	LU	Correo electrónico
Lopez Valiente, Patricio	457/15	patriciolopezvaliente@gmail.com

Reservado para la catedra

Instancia	Docente	Nota
Primera entrega		
Segunda entrega		

Índice

1. Introducción 3

2. Desarrollo 3

2.1. Técnicas utilizadas . . . 3

2.1.1. BackPropagation . . . 3

2.2. Implementación . . . 4

3. Experimentos 4

3.1. . . . 4

3.2. Detección de Cáncer de Mamas . . . 4

3.3. Eficiencia Energética . . . 6

4. Conclusiones 8

1. Introducción

En este trabajo presentaremos una técnica de aprendizaje supervisado ampliamente conocida como BackPropagation Esta técnica es muy útil a la hora de abordar problemas complejo, ya que permiten resolver problemas complejos, a partir de mediciones reales del problema. Ademas como es una técnica de Aprendizaje supervisado, la dificultad del problema a resolver se traslada al modelado de la Red Neuronal a usar, por lo que tiene especial utilidad cuando no sabemos como abordar el problema a tratar.

El objetivo principal de este informe es analizar y presentar el Modelado de Redes Neuronales utilizando el algoritmo de BackPropagation. Para mostrar las capacidades, ventajas y desventajas de esta técnicas resolvieron los siguientes problemas, los cuales fueron propuestos por la cátedra:

- **Detección de Cáncer de Mamas:** Este problema consiste en decidir si datos referentes a imágenes medicas, son indicativos de un tumor maligno o benigno. Para esto se presenta un conjunto de datos, el cual contiene los resultados de un examen específico que es utilizado en el diagnóstico del cáncer de mamas. Cada entrada corresponde a los datos obtenidos para distintos pacientes y contiene 10 características provenientes de imágenes digitalizadas de muestras de células. Ademas para realizar el Aprendizaje supervisado junto con estas características se encuentra también el diagnóstico final, determinado junto con otras pruebas, en donde se indica si la muestra analizada pertenecía a un tumor maligno o benigno.
- **Eficiencia Energética:** Este problema consiste en determinar los requerimientos de carga energética para calefaccionar y refrigerar edificios en función de ciertas características de los mismos. El análisis energético se realizó utilizando edificios de distintas formas que difieren con respecto a la superficie y distribución de las áreas de reflejo, la orientación y otros parámetros. Cada entrada en el conjunto de datos corresponde a las características de un edificio distinto junto a dos valores reales que representan la cantidad de energía necesaria para realizar una calefacción y refrigeración adecuadas, con los cuales se realiza el Aprendizaje supervisado.

2. Desarrollo

2.1. Técnicas utilizadas

2.1.1. BackPropagation

Es el algoritmo de Aprendizaje supervisado mas utilizado, dado que es muy simple de programar y también es lo suficientemente robusto como para poder dar solución a problemas muy complejos.
La aplicación del algoritmo tiene dos fases, una hacia delante y otra hacia atrás. Durante la primera fase el patrón de entrada es presentado a la red y propagado a

través de las capas hasta llegar a la capa de salida. Obtenidos los valores de salida de la red, se inicia la segunda fase, comparándose éstos valores con la salida esperada para así obtener el error. Se ajustan los pesos de la última capa proporcionalmente al error. Se pasa a la capa anterior con una retropopagación del error, ajustando los pesos y continuando con este proceso hasta llegar a la primera capa. De esta manera se han modificado los pesos de las conexiones de la red para cada patrón de aprendizaje del problema, del que conocíamos su valor de entrada y la salida deseada que debería generar la red ante dicho patrón.
La técnica Backpropagation requiere el uso de neuronas cuya función de activación sea continua, y por lo tanto, diferenciable. Generalmente, la función utilizada es la sigmoide. Aunque en este trabajo utilizaremos la tangente hiperbólica, ya que presento mejor desempeño en las pruebas iniciales.

2.2. Implementación

La implementación de los Algoritmos realizados fue hecha en python, el ejecutable generado se puede utilizar de forma interactiva de manera simple, este esta preparado para utilizar bases de datos csv con el formato provisto por la cátedra en ambos ejercicios. El ejecutable se utiliza mediante inputs, los cuales interactuan con el usuario, solicitándole los datos de prueba y demás parámetros. La implementación de los algoritmos utilizados, no va a ser desarrollada en este informe, así como tampoco se analizara su complejidad, ya que si bien son cualidades computacionales importantes, no radican importancia con respecto al modelado de Redes Neuronales.

3. Experimentos

3.1.

Para la fase de experimentación se utilizaron todos los datos provistos por la cátedra, de los cuales el 20 % se los utilizo para validar la fase de training de la Red. Ademas se utilizo como alpha inicial y control $\alpha = 0{,}5$.

3.2. Detección de Cáncer de Mamas

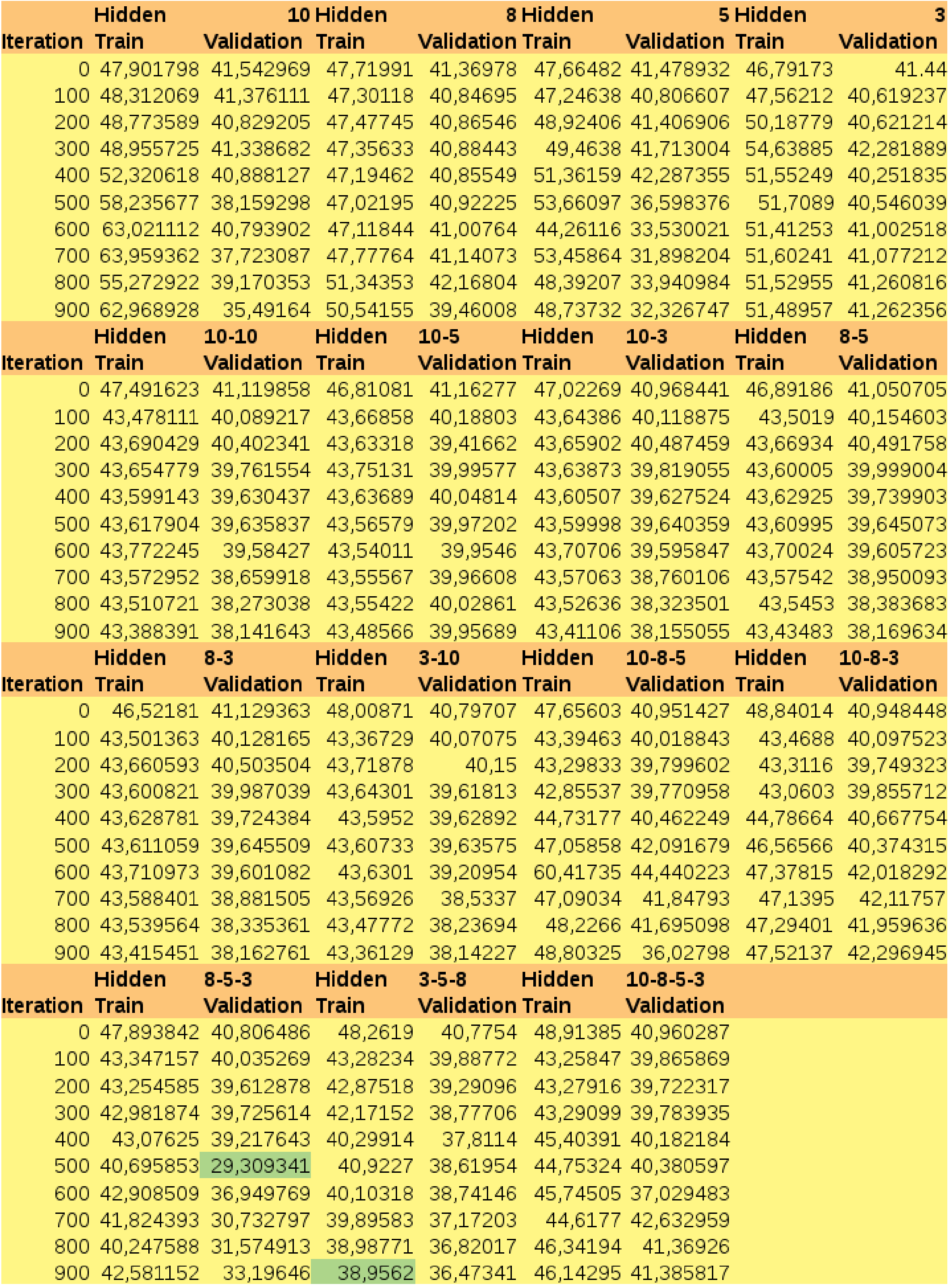
Para este problema se experimento con redes con entrada 10 entradas, una salida y las siguientes configuraciones de capas intermedias:

- Una capa con: 10, 8, 5 o 3 neuronas.
- Dos capas con: (10, 10), (10, 5), (10, 3), (8, 5), (8, 3) o (3, 10) neuronas.
- Tres capas con: (10, 8, 5), (10, 8, 3),(8, 5, 5) o (3, 5, 8) neuronas.
- Cuatro capas con (10, 8, 5, 3) neuronas.

Se utilizo el alpha control, y se midió el error sobre el conjunto de training y de test cada 100 iteraciones, desde la inicial, hasta la 900.

los resultados obtenidos fueron los siguientes:

Figura 1. Error - $\alpha(0.5)$ - Cáncer de Mamas

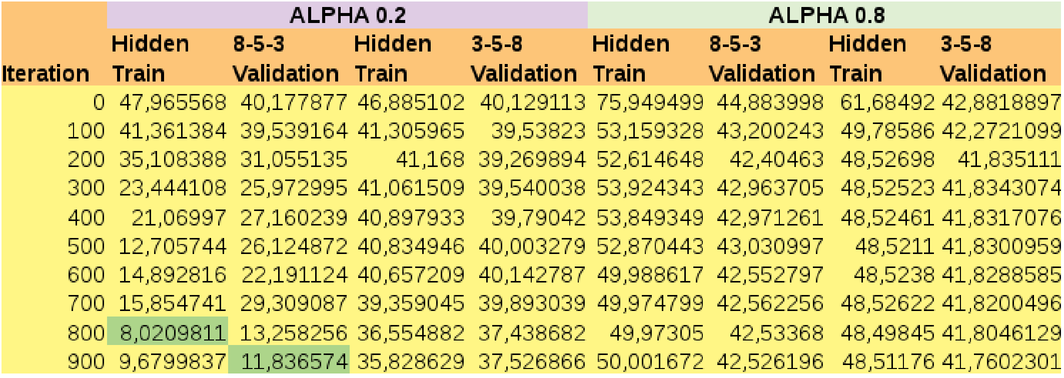


	Hidden	10	Hidden	8	Hidden	5	Hidden	3
Iteration	Train	Validation	Train	Validation	Train	Validation	Train	Validation
0	47,901798	41,542969	47,71991	41,36978	47,66482	41,478932	46,79173	41.44
100	48,312069	41,376111	47,30118	40,84695	47,24638	40,806607	47,56212	40,619237
200	48,773589	40,829205	47,47745	40,86546	48,92406	41,406906	50,18779	40,621214
300	48,955725	41,338682	47,35633	40,88443	49,4638	41,713004	54,63885	42,281889
400	52,320618	40,888127	47,19462	40,85549	51,36159	42,287355	51,55249	40,251835
500	58,235677	38,159298	47,02195	40,92225	53,66097	36,598376	51,7089	40,546039
600	63,021112	40,793902	47,11844	41,00764	44,26116	33,530021	51,41253	41,002518
700	63,959362	37,723087	47,77764	41,14073	53,45864	31,898204	51,60241	41,077212
800	55,272922	39,170353	51,34353	42,16804	48,39207	33,940984	51,52955	41,260816
900	62,968928	35,49164	50,54155	39,46008	48,73732	32,326747	51,48957	41,262356
	Hidden	10-10	Hidden	10-5	Hidden	10-3	Hidden	8-5
Iteration	Train	Validation	Train	Validation	Train	Validation	Train	Validation
0	47,491623	41,119858	46,81081	41,16277	47,02269	40,968441	46,89186	41,050705
100	43,478111	40,089217	43,66858	40,18803	43,64386	40,118875	43,5019	40,154603
200	43,690429	40,402341	43,63318	39,41662	43,65902	40,487459	43,66934	40,491758
300	43,654779	39,761554	43,75131	39,99577	43,63873	39,819055	43,60005	39,999004
400	43,599143	39,630437	43,63689	40,04814	43,60507	39,627524	43,62925	39,739903
500	43,617904	39,635837	43,56579	39,97202	43,59998	39,640359	43,60995	39,645073
600	43,772245	39,58427	43,54011	39,9546	43,70706	39,595847	43,70024	39,605723
700	43,572952	38,659918	43,55567	39,96608	43,57063	38,760106	43,57542	38,950093
800	43,510721	38,273038	43,55422	40,02861	43,52636	38,323501	43,5453	38,383683
900	43,388391	38,141643	43,48566	39,95689	43,41106	38,155055	43,43483	38,169634
	Hidden	8-3	Hidden	3-10	Hidden	10-8-5	Hidden	10-8-3
Iteration	Train	Validation	Train	Validation	Train	Validation	Train	Validation
0	46,52181	41,129363	48,00871	40,79707	47,65603	40,951427	48,84014	40,948448
100	43,501363	40,128165	43,36729	40,07075	43,39463	40,018843	43,4688	40,097523
200	43,660593	40,503504	43,71878	40,15	43,29833	39,799602	43,3116	39,749323
300	43,600821	39,987039	43,64301	39,61813	42,85537	39,770958	43,0603	39,855712
400	43,628781	39,724384	43,5952	39,62892	44,73177	40,462249	44,78664	40,667754
500	43,611059	39,645509	43,60733	39,63575	47,05858	42,091679	46,56566	40,374315
600	43,710973	39,601082	43,6301	39,20954	60,41735	44,440223	47,37815	42,018292
700	43,588401	38,881505	43,56926	38,5337	47,09034	41,84793	47,1395	42,11757
800	43,539564	38,335361	43,47772	38,23694	48,2266	41,695098	47,29401	41,959636
900	43,415451	38,162761	43,36129	38,14227	48,80325	36,02798	47,52137	42,296945
	Hidden	8-5-3	Hidden	3-5-8	Hidden	10-8-5-3		
Iteration	Train	Validation	Train	Validation	Train	Validation		
0	47,893842	40,806486	48,2619	40,7754	48,91385	40,960287		
100	43,347157	40,035269	43,28234	39,88772	43,25847	39,865869		
200	43,254585	39,612878	42,87518	39,29096	43,27916	39,722317		
300	42,981874	39,725614	42,17152	38,77706	43,29099	39,783935		
400	43,07625	39,217643	40,29914	37,8114	45,40391	40,182184		
500	40,695853	29,309341	40,9227	38,61954	44,75324	40,380597		
600	42,908509	36,949769	40,10318	38,74146	45,74505	37,029483		
700	41,824393	30,732797	39,89583	37,17203	44,6177	42,632959		
800	40,247588	31,574913	38,98771	36,82017	46,34194	41,36926		
900	42,581152	33,19646	38,9562	36,47341	46,14295	41,385817		

En la Figura Figura 1 vemos en verde los errores mínimos obtenidos tanto para la validación, como para el training, estos se obtienen respectivamente en las configuraciones 8-5-3 y 3-5-8. También en la Figura Figura 1 vemos que no se obtuvo un error pequeño, por lo que se podría decir que no se obtuvo una gran configuración, pero teniendo en cuenta que la salida con la que se chequea el resultado es discreta(0 o 1), se entiende que esta va a obtener mayor error que si fuera continua y tomara un umbral de valores.

Ahora habiendo obtenido las dos mejores configuraciones, se puso a prueba el factor de aprendizaje, con el fin de obtener una configuración optima, o cercana a esta.

Figura 2. Error - α variable - Cáncer de Mamas



	ALPHA 0.2				ALPHA 0.8			
	Hidden	8-5-3	Hidden	3-5-8	Hidden	8-5-3	Hidden	3-5-8
Iteration	Train	Validation	Train	Validation	Train	Validation	Train	Validation
0	47,965568	40,177877	46,885102	40,129113	75,949499	44,883998	61,68492	42,8818897
100	41,361384	39,539164	41,305965	39,53823	53,159328	43,200243	49,78586	42,2721099
200	35,108388	31,055135	41,168	39,269894	52,614648	42,40463	48,52698	41,835111
300	23,444108	25,972995	41,061509	39,540038	53,924343	42,963705	48,52523	41,8343074
400	21,06997	27,160239	40,897933	39,79042	53,849349	42,971261	48,52461	41,8317076
500	12,705744	26,124872	40,834946	40,003279	52,870443	43,030997	48,5211	41,8300959
600	14,892816	22,191124	40,657209	40,142787	49,988617	42,552797	48,5238	41,8288585
700	15,854741	29,309087	39,359045	39,893039	49,974799	42,562256	48,52622	41,8200496
800	8,0209811	13,258256	36,554882	37,438682	49,97305	42,53368	48,49845	41,8046129
900	9,6799837	11,836574	35,828629	37,526866	50,001672	42,526196	48,51176	41,7602301

En la Figura Figura 2 vemos como utilizando $\alpha = 0{,}2$ obtenemos una configuración mucho mas eficiente. Esta vez coincidiendo ambos mínimos en la configuración 8-5-3, aunque con distintos números de iteraciones, lo que podría indicar que entre 800 y 900 se encuentra un mínimo local.

Dado los experimentos realizados se tomo como configuración optima utilizar: 3 capas ocultas con (8, 5, 3) neuronas, $\alpha = 0{,}2$ y 800 iteraciones, ya que esta es la que minimizo el error.

3.3. Eficiencia Energética

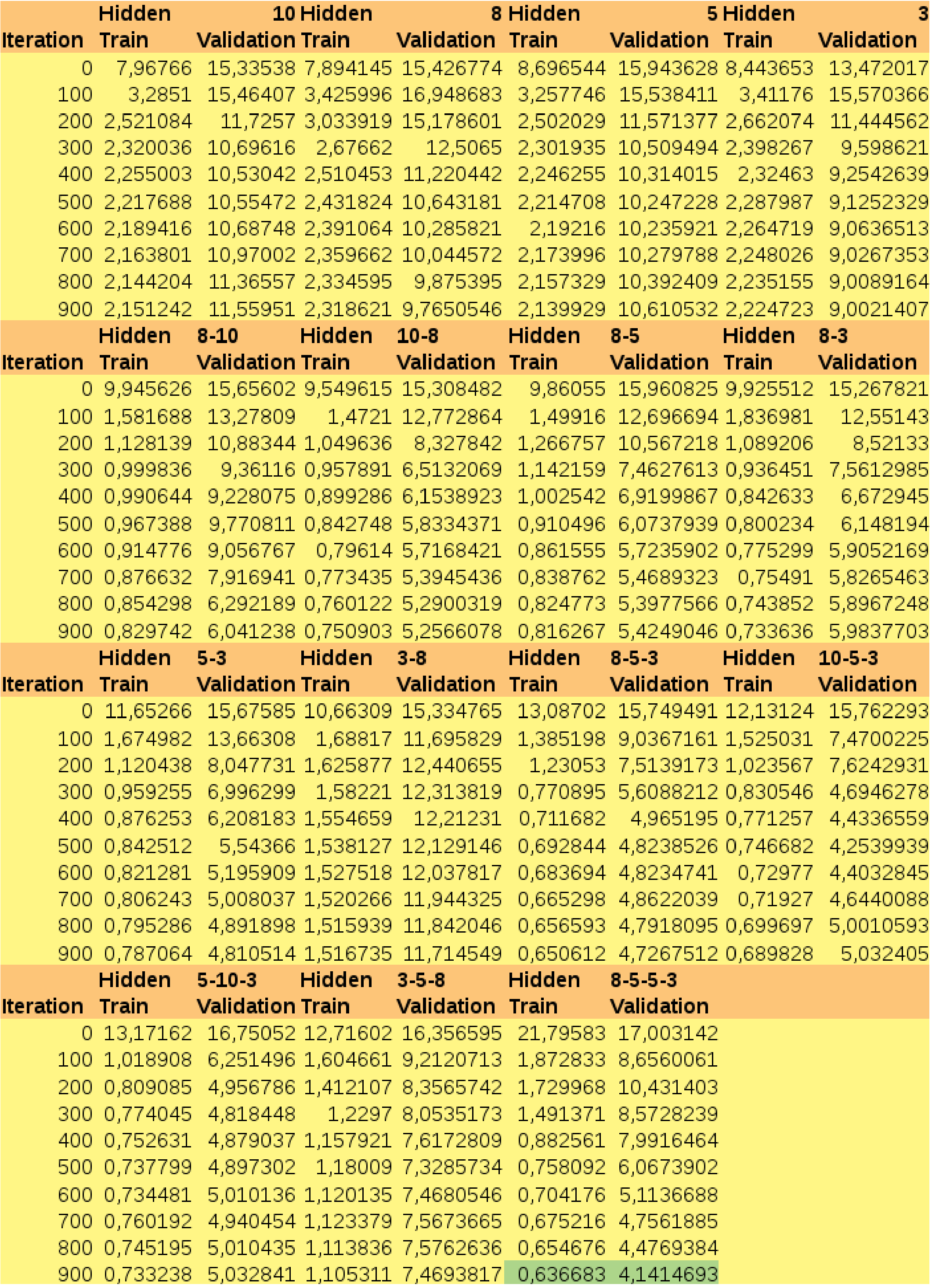
Para este problema se experimento con redes con entrada 8 entradas, 2 salidas y las siguientes configuraciones de capas intermedias:

- Una capa con: 10, 8, 5 o 3 neuronas.
- Dos capas con: (10, 8), (8, 10), (8, 5), (8, 3), (5, 3) o (3, 8) neuronas.
- Tres capas con: (8, 5, 3), (10, 5, 3),(5, 10, 3) o (3, 5, 8) neuronas.
- Cuatro capas con (8, 5, 5, 3) neuronas.

Se utilizo el alpha control, y se midió el error sobre el conjunto de training y de test cada 100 iteraciones, desde la inicial, hasta la 900.

los resultados obtenidos fueron los siguientes:

Figura 3. Error - $\alpha(0.5)$ - Eficiencia Energética

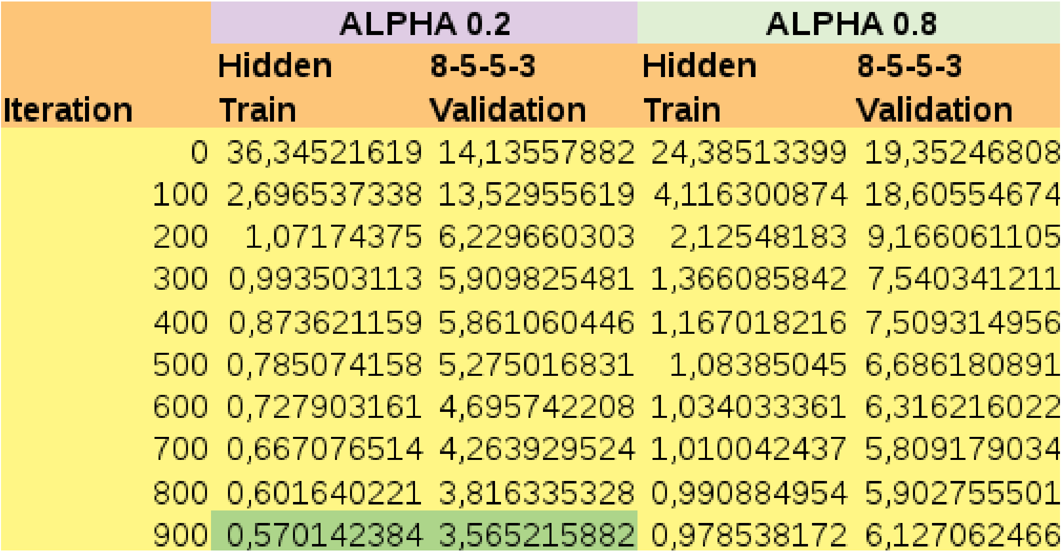


	Hidden	10	Hidden	8	Hidden	5	Hidden	3
Iteration	**Train**	**Validation**	**Train**	**Validation**	**Train**	**Validation**	**Train**	**Validation**
0	7,96766	15,33538	7,894145	15,426774	8,696544	15,943628	8,443653	13,472017
100	3,2851	15,46407	3,425996	16,948683	3,257746	15,538411	3,41176	15,570366
200	2,521084	11,7257	3,033919	15,178601	2,502029	11,571377	2,662074	11,444562
300	2,320036	10,69616	2,67662	12,5065	2,301935	10,509494	2,398267	9,598621
400	2,255003	10,53042	2,510453	11,220442	2,246255	10,314015	2,32463	9,2542639
500	2,217688	10,55472	2,431824	10,643181	2,214708	10,247228	2,287987	9,1252329
600	2,189416	10,68748	2,391064	10,285821	2,19216	10,235921	2,264719	9,0636513
700	2,163801	10,97002	2,359662	10,044572	2,173996	10,279788	2,248026	9,0267353
800	2,144204	11,36557	2,334595	9,875395	2,157329	10,392409	2,235155	9,0089164
900	2,151242	11,55951	2,318621	9,7650546	2,139929	10,610532	2,224723	9,0021407
	Hidden	**8-10**	**Hidden**	**10-8**	**Hidden**	**8-5**	**Hidden**	**8-3**
Iteration	**Train**	**Validation**	**Train**	**Validation**	**Train**	**Validation**	**Train**	**Validation**
0	9,945626	15,65602	9,549615	15,308482	9,86055	15,960825	9,925512	15,267821
100	1,581688	13,27809	1,4721	12,772864	1,49916	12,696694	1,836981	12,55143
200	1,128139	10,88344	1,049636	8,327842	1,266757	10,567218	1,089206	8,52133
300	0,999836	9,36116	0,957891	6,5132069	1,142159	7,4627613	0,936451	7,5612985
400	0,990644	9,228075	0,899286	6,1538923	1,002542	6,9199867	0,842633	6,672945
500	0,967388	9,770811	0,842748	5,8334371	0,910496	6,0737939	0,800234	6,148194
600	0,914776	9,056767	0,79614	5,7168421	0,861555	5,7235902	0,775299	5,9052169
700	0,876632	7,916941	0,773435	5,3945436	0,838762	5,4689323	0,75491	5,8265463
800	0,854298	6,292189	0,760122	5,2900319	0,824773	5,3977566	0,743852	5,8967248
900	0,829742	6,041238	0,750903	5,2566078	0,816267	5,4249046	0,733636	5,9837703
	Hidden	**5-3**	**Hidden**	**3-8**	**Hidden**	**8-5-3**	**Hidden**	**10-5-3**
Iteration	**Train**	**Validation**	**Train**	**Validation**	**Train**	**Validation**	**Train**	**Validation**
0	11,65266	15,67585	10,66309	15,334765	13,08702	15,749491	12,13124	15,762293
100	1,674982	13,66308	1,68817	11,695829	1,385198	9,0367161	1,525031	7,4700225
200	1,120438	8,047731	1,625877	12,440655	1,23053	7,5139173	1,023567	7,6242931
300	0,959255	6,996299	1,58221	12,313819	0,770895	5,6088212	0,830546	4,6946278
400	0,876253	6,208183	1,554659	12,21231	0,711682	4,965195	0,771257	4,4336559
500	0,842512	5,54366	1,538127	12,129146	0,692844	4,8238526	0,746682	4,2539939
600	0,821281	5,195909	1,527518	12,037817	0,683694	4,8234741	0,72977	4,4032845
700	0,806243	5,008037	1,520266	11,944325	0,665298	4,8622039	0,71927	4,6440088
800	0,795286	4,891898	1,515939	11,842046	0,656593	4,7918095	0,699697	5,0010593
900	0,787064	4,810514	1,516735	11,714549	0,650612	4,7267512	0,689828	5,032405
	Hidden	**5-10-3**	**Hidden**	**3-5-8**	**Hidden**	**8-5-5-3**		
Iteration	**Train**	**Validation**	**Train**	**Validation**	**Train**	**Validation**		
0	13,17162	16,75052	12,71602	16,356595	21,79583	17,003142		
100	1,018908	6,251496	1,604661	9,2120713	1,872833	8,6560061		
200	0,809085	4,956786	1,412107	8,3565742	1,729968	10,431403		
300	0,774045	4,818448	1,2297	8,0535173	1,491371	8,5728239		
400	0,752631	4,879037	1,157921	7,6172809	0,882561	7,9916464		
500	0,737799	4,897302	1,18009	7,3285734	0,758092	6,0673902		
600	0,734481	5,010136	1,120135	7,4680546	0,704176	5,1136688		
700	0,760192	4,940454	1,123379	7,5673665	0,675216	4,7561885		
800	0,745195	5,010435	1,113836	7,5762636	0,654676	4,4769384		
900	0,733238	5,032841	1,105311	7,4693817	0,636683	4,1414693		

En la Figura Figura 3 vemos en verde los errores mínimos obtenidos tanto para la validación, como para el training, estos en este caso coinciden, en la misma cantidad de iteraciones en la configuración 8-5-5-3. En este caso a diferencia del anterior si se obtuvo un error relativamente pequeño error pequeño, quizás esto reafirmaría la hipótesis sobre la continuidad de la salida en problemas complejos, ya que este modelo de Red podría fácilmente calcular problemas estrictamente discretos como XOR, y problemas similares.

Utilizando la configuración que minimiza el error, se procedió a evaluar el factor de aprendizaje de esta configuración.

Figura 4. Error - α variable - Eficiencia Energética



	ALPHA 0.2		ALPHA 0.8	
	Hidden	8-5-5-3	Hidden	8-5-5-3
Iteration	Train	Validation	Train	Validation
0	36,34521619	14,13557882	24,38513399	19,35246808
100	2,696537338	13,52955619	4,116300874	18,60554674
200	1,07174375	6,229660303	2,12548183	9,166061105
300	0,993503113	5,909825481	1,366085842	7,540341211
400	0,873621159	5,861060446	1,167018216	7,509314956
500	0,785074158	5,275016831	1,08385045	6,686180891
600	0,727903161	4,695742208	1,034033361	6,316216022
700	0,667076514	4,263929524	1,010042437	5,809179034
800	0,601640221	3,816335328	0,990884954	5,902755501
900	0,570142384	3,565215882	0,978538172	6,127062466

En la Figura Figura 4 vemos como utilizando $\alpha = 0{,}2$ nuevamente obtenemos una configuracion mas eficiente, aunque en este caso la diferencia es mínima, de orden decimal. Nuevamente coincidiendo ambos mínimos en la configuracion 8-5-5-3, con igual numero de iteraciones, también vemos como un factor de aprendizaje alto perjudica el aprendizaje de la red para un numero alto de iteraciones.

Dado los experimentos realizados se tomo como configuracion optima utilizar: 4 capas ocultas con (8, 5, 5, 3) neuronas, $\alpha = 0{,}2$ y 900 iteraciones, ya que esta es la que minimizo el error.

4. Conclusiones

Las conclusiones obtenidas luego de realizar este informe y de haber llevado a cabo los experimentos, son:

1. Es conveniente usar un factor de Aprendizaje pequeño, o en una opción superadora variable decreciente, ya que a medida que iteramos la red requiere calibres cada vez mas finos, que un factor grande no puede brindar, por lo que es preferible realizar el tradeOff con el tiempo la convergencia de la Red, en pos de una mayor eficiencia de esta.

2. Pareciera que es mas fácil aproximar funciones, de una complejidad alta, que posean imagen continua, que las que poseen imagen discreta. Como Ocurre con el Diagnostico de Cáncer de Mamas.

3. Las Redes neuronales son una gran herramienta a la hora de hallar soluciones a problemas complejos, pero el modelado de las mismas no es nada sencillo, y pareciera ser que no hay otra forma de encontrar una eficiente, mas que realizando pruebas y experimentos.