Redes Neurais - Backpropagation
Trabalho 2 – INF01017/CMP263 – 2018/2

Cristian Leal Nornberg[1] , Filipe Lins[1], Pablo Pavan[1]

[1]Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

clnornberg@inf.ufrgs.br, fmlins@inf.ufrgs.br, pjpavan@inf.ufrgs.br

1. Introdução

Redes neurais artificiais são modelos preditivos baseados no funcionamento dos neurônios do cérebro. Cada neurônio recebe sinais (entradas) provenientes de outros neurônios que o alimentam, faz um cálculo e então sofre ativação se o cálculo exceder um limite ou não. Redes neurais são formadas por neurônios artificiais que desenvolvem cálculos a partir de suas entradas. Podem resolver diversos problemas como reconhecimento de imagens, processamento de linguagem natural, reconhecimento de áudio, entre outros. As redes neurais podem ser divididas em recorrentes (com retroalimentação) e redes de alimentação direta (feedforward).

Este trabalho tem o objetivo de implementar uma rede neural feedforward, com número ajustável de neurônios, treinada a partir de diferentes conjuntos de dados, e utiliza o algoritmo de backpropagation para a correção do erro da rede. A linguagem de programação utilizada foi Python 3. Foram implementadas funções que permitem realizar a verificação numérica do gradiente, um mecanismo para normalização dos dados de treinamento, assim como uso de regularização.

2. Redes Neurais Artificiais

Redes neurais artificiais utilizam neurônios matemáticos, que são modelos simplificados que fazem analogia aos neurônios do cérebro humano, inspirados na geração e propagação de impulsos elétricos pela membrana celular dos neurônios. As redes neurais são formadas por camadas (layers), onde a primeira camada é chamada de camada de entrada (input layer), a última camada é chamada de camada de saída (output layer), e uma ou mais camadas intermediárias que são chamadas de camadas ocultas (hidden layers). Cada neurônio está interligado com todos os neurônios da camada seguinte. A informação é passada através de cada camada, e a saída da camada anterior serve como entrada da próxima camada. A força de conexão entre dois neurônios é determinada pelo peso sináptico de cada ligação. As saídas de cada camada utilizam algoritmos contendo uma função de ativação.

Cada neurônio calcula primeiro a soma ponderada de suas entradas, onde as entradas são multiplicadas pelos seus respectivos pesos sinápticos e então somados. Esta função é denominada função de combinação. Os pesos podem ser positivos ou negativos, dependendo de o comportamento da conexão ser excitatório ou inibitório. Um valor de peso zero significa que não há conexão associada. A cada neurônio é adicionado um termo de desvio ou termo de bias. O termo de bias permite que um neurônio apresente uma saída não-nula mesmo que todas suas entradas sejam nulas, aumentando o grau de independência da função. O valor de bias é adicionado ao somatório da função de ativação. Este resultado é então apresentado à uma função de ativação, de onde resulta o valor de saída.

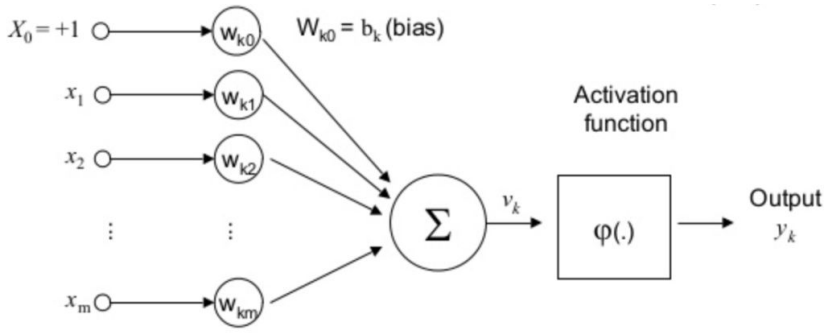

Figura 1. Função de ativação

Várias funções de ativação têm sido propostas, como função linear, sigmóide, tangente hiperbólica, ReLU, entre outras. Neste trabalho optamos por utilizar a função sigmóide, também conhecida como função logística, por ser uma função suave e continuamente diferenciável, o que é importante para o algoritmo de aprendizagem dos pesos. O resultado da função varia de 0 à 1 e seu gráfico tem formato de S.

As redes neurais podem ser divididas em duas categorias, de acordo com seu grau de conectividade: redes de alimentação direta (feedforward) e redes recorrentes. Sua diferença está baseada na presença ou não de retroalimentação (feedback). Nas redes neurais feedforward a informação flui em uma única direção: da entrada da rede, passando pelos neurônios das camadas ocultas, para a camada de saída. Nas redes recorrentes existem conexões de retroalimentação, que permitem que os neurônios da mesma camada recebam em seus terminais de entrada a saída de um neurônio da mesma camada, de uma camada posterior, ou até mesmo a sua própria saída. Neste trabalho foram utilizadas redes neurais feedforward.

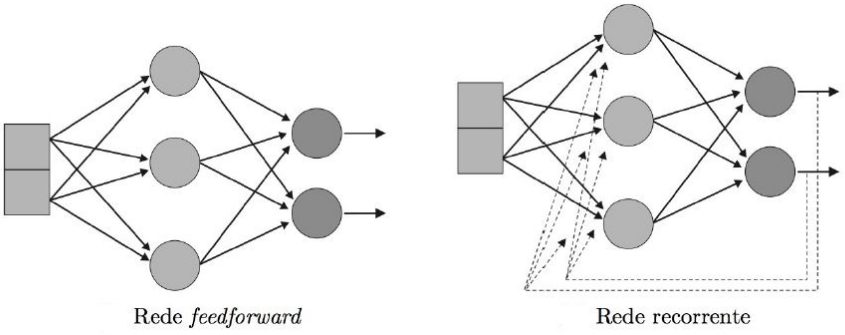

Figura 2. Rede Feedforward x Rede Recorrente

3. Aprendizado da Rede Neural

O objetivo dos algoritmos de aprendizagem de redes neurais é ajustar os pesos da rede, diminuindo assim os erros cometidos pela rede, ou seja, encontrar um conjunto de pesos que tornem o erro o menor possível. Em uma rede neural multicamadas, cada neurônio realiza uma função específica. A função de um neurônio de uma dada camada é a combinação das funções realizadas pelos neurônios da camada anterior. Cada neurônio da camada de saída está associado à uma das classes do conjunto de dados. Durante o processo de treinamento da rede, o vetor de respostas desejadas possui valor 1 para a classe correta da instância e 0 (zero) para as demais classes. O erro é calculado pela comparação entre o vetor de saída dos neurônios da camada de saída e o vetor de valores desejados. A rede classifica corretamente quando o valor mais elevado produzido é gerado pelo neurônio de saída correspondente à classe correta, e um erro de classificação ocorre quando o

neurônio de uma outra classe produz o valor de saída mais elevado.

Para a correção dos erros da rede neural, foi utilizado o algoritmo de backpropagation, baseado em gradiente descendente. O backpropagation possui duas fases:

Para frente (forward): Cada instância do conjunto de treinamento é recebido pela camada de entrada e passado para cada um dos neurônios da primeira camada, onde é ponderado pelos pesos associado a suas conexões de entrada. Então cada neurônio aplica a função de ativação e produz um valor de saída que é utilizado como entrada pelos neurônios da camada seguinte. Esse processo continua até que os neurônios da camada de saída produzam seu valor de saída, que é então comparado com o valor desejado para a saída deste neurônio. A diferença entre os valores de saída e o desejado indica o erro da rede para a instância apresentada.

Para trás (backward): O valor de erro de cada neurônio da camada de saída é utilizado na fase backward, para ajustar os pesos. O ajuste segue da camada de saída até a primeira camada oculta. Como os valores dos erros são conhecidos apenas para os neurônios da camada de saída (erro na camada de saída é igual ao erro da rede), o erro para as camadas intermediárias precisa ser estimado.

No backpropagation, o erro de um neurônio das camadas intermediárias é estimado calculando-se a soma dos erros dos neurônios da camada seguinte, cujos terminais de entrada estão ligados a ele, ponderados pelo peso associado a essas conexões e a ativação do neurônio. O resultado indica o quanto a saída de um neurônio contribuiu para o erro da rede, e este valor será utilizado no cálculo para a correção dos pesos.

$$\delta_i^{l=k} = \left(\sum_{j=1}^{N} \theta_{ij}^{l=k+1} \ \delta_j^{l=k+1} \right) \left(a_i^{l=k} \right) \left(1 - a_i^{l=k} \right)$$

Figura 3. Cálculo do erro estimado para neurônios das camadas internas.

Para que a correção dos erros seja realizada de forma gradual, evitando passos grandes, é introduzida no cálculo de atualização dos pesos a taxa de aprendizagem. A taxa de aprendizagem é uma constante, que tem forte influência no tempo necessário para convergência da rede. Se for utilizada uma taxa muito pequena, podem ser necessários muitos ciclos até a convergência, porém se utilizada uma taxa muito alta, podem ocorrer oscilações que dificultam a convergência.

Dados contendo atributos com escalas de valores diferentes podem afetar o treinamento dos modelos. Para evitar que um atributo predomine sobre outro, é necessário realizar a normalização dos dados. Para isso, utilizamos a normalização por reescala, também chamada de min-max. Primeiro o valor mínimo do atributo é subtraído do valor atual, e dividido pela diferença entre o maior valor do atributo e o menor valor do atributo.

$$x_j^{norm} = \frac{x_j - x_j^{\min}}{x_j^{\max} - x_i^{\min}}$$

Figura 4. Normalização de dados min-max

Redes com muitas camadas e/ou neurônios nem sempre são as melhores opções, pois podem sofrer problemas de overfitting, devido à complexidade da rede. Overfitting ocorre quando o modelo é capaz de acertar todos dados de treinamento, porém falha para dados novos. Para reduzir o overfitting, podem ser utilizadas técnicas de regularização. A técnica utilizada neste trabalho foi a técnica de decaimento dos pesos (weight decay), onde uma constante chamada termo de

regularização é adicionada à função de cálculo do custo. O objetivo é fazer com que a rede aprenda pequenos pesos, só permitindo pesos maiores caso melhorem a função de custo. Se a taxa de regularização for 0 (zero), nenhuma feature é penalizada, o que pode tornar o modelo muito complexo, causando overfitting. Se a taxa de regularização for muito alta, para minimizar a função custo todos os pesos são forçados a ter valor 0 (zero), tornando o modelo muito simples, causando underfitting.

A função de custo no caso do gradiente descendente tem como objetivo encontrar o melhor ajuste para determinado conjunto de entradas e saídas nos mostrando um valor real que representa o custo associado a um determinado evento. Dessa maneira estamos associando a saídas esperadas em cada neurônio e o valor atual encontrado no cálculo da função de custo. Neste trabalho a função de custo utilizada foi a entropia cruzada:

$$J(\theta_{11}^{l=1}, \theta_{12}^{l=1}, \ldots, \theta_{11}^{l=L}, \ldots) = \frac{1}{n}\sum_{i=1}^{n} -y^{(i)}\Big(\ \log(f_\theta(x^{(i)})\Big) - (1-y^{(i)})\Big(\ \log(1-f_\theta(x^{(i)}))\Big)$$

Figura 5. Cálculo da Função de Custo.

4. Conjuntos de Dados

Os seguintes conjuntos de dados foram utilizados para o treinamento das redes neurais:

A. Pima Indian Diabetes Data Set: Conjunto de dados formado por 8 atributos, 768 instâncias e 2 classes, onde o objetivo é predizer se um paciente tem diabetes a partir de um pequeno conjunto de dados clínicos.
B. Wine Data Set: Conjunto de dados formado por 13 atributos, 178 instâncias e 3 classes, onde o objetivo é predizer o tipo de vinho baseado em sua composição química.
C. Ionosphere Data Set: Conjunto de dados formado por 34 atributos, 351 instâncias e 2 classes, onde o objetivo é predizer se a captura de sinais de um radar da ionosfera é adequada para análises posteriores.
D. Breast Cancer Wisconsin: Conjunto de dados formado por 32 atributos, 569 instâncias e classes, onde o objetivo é predizer se um exame médico indica ou não a presença de câncer.

5. Implementação

Para o desenvolvimento dos programas, foi utilizada a linguagem Python 3.6, com os bibliotecas numpy, argcomplete e matplotlib e a função shuffle da biblioteca sklearn.utils, esta última somente nas implementações para a validação dos datasets. Todos os códigos e resultados estão disponíveis no github no seguinte link: https://github.com/PabloPavan/machine-learning

5.1 Benchmark de Validação

O benchmark para validar a implementação do backpropagation, está na pasta benchmark e para ser executado basta chama-lo com a seguinte linha de comando: *python3 main_example.py --n network.txt --w initial_weights.txt --d dataset.txt* onde, *--n* recebe o arquivo de configuração da rede, *--w* recebe os pesos iniciais e *--d* o dataset a ser testado. A saída do benchmark é um print na tela com o mesmo formato do arquivo de exemplo passado pelo professor.

5.2 Experimentos

Para cada dataset foi criada um arquivo main que recebe como parâmetro o arquivo de network a ser executado. Este arquivo contém na primeira linha o valor da taxa de regularização (lambda), na segunda linha o valor da taxa de aprendizagem (alpha), o número de entradas da rede na terceira linha, as linhas seguintes contém o número de neurônios nas camadas ocultas e a última linha representa o número de saídas. A inicialização dos pesos é realizada pela função build_weights encontrada em utils.py, que é executada para cada execução do k-fold.

As redes neurais foram treinadas utilizando diferentes configurações de rede, variando-se o número de camadas e neurônios. Além disso, foram testados diferentes valores para a taxa de aprendizagem (alpha) e taxa de regularização (lambda). O número de camadas de rede utilizados variou de 1 à 4, e o número de neurônios em cada camada utilizados nos testes foram 2, 4, 8 e 16. Os valores utilizados para alpha foram 0,1 e 0,01. Os valores utilizados para lambda foram 0, 0,1 e 0,001. As redes neurais foram treinadas utilizando diferentes conjuntos de dados, onde foi avaliada a performance da rede utilizando-se F1-Measure.

Para o plot do J foi criado uma outra aplicação chamada main_plot.py que recebe por parâmetro o arquivo network, o dataset, e o nome da figura a ser gerada. Um exemplo de execução: *python3 main_plot.py net_beast_ionosphere.txt data/ionosphere.data ionosphere_J.png*.

Foram realizados 10 repetições de cada configuração para posteriormente realizar a média das execuções, foi coletado também o tempo de execução para cada configuração. Os testes foram executados sobre as máquinas do parque computacional do GPPD-HPC (http://gppd-hpc.inf.ufrgs.br/). Todos as análises foram feitas utilizando a linguagem R, juntamente com a biblioteca ggplot2 para a geração dos gráficos. Somente os gráficos do resultado do J foram feitos utilizando a biblioteca matplotlib durante a própria execução.

Um projeto experimental de 2k é usado para determinar o efeito de fatores k, cada um com duas alternativas ou níveis. Essa classe de planejamento fatorial merece uma discussão especial, pois é fácil de analisar e ajuda na classificação de fatores na ordem do impacto. No início de um estudo de desempenho, o número de fatores e seus níveis costumam ser grandes. Um planejamento fatorial completo com um número tão grande de fatores e níveis pode não ser o melhor uso do esforço disponível. O primeiro passo deve ser reduzir o número de fatores e escolher aqueles fatores que têm impacto significativo no desempenho. Muitas vezes, o efeito de um fator é unidirecional, ou seja, o desempenho diminui continuamente ou aumenta continuamente à medida que o fator é aumentado de mínimo para máximo. A porcentagem de variação ajuda o experimentador a decidir se vale a pena investigar mais um fator ou interação (JAIN, Raj. 1990).

Nos nossos experimentos optamos por utilizar um projeto 2k usando K= 4 fatores, onde os fatores são, o número de camadas ocultas (A), o número de neurônios por camada (B), a regularização (C) e a taxa de aprendizado (D). A tabela X apresenta os resultados das porcentagens de variação, indicando o quão importante no desempenho da F1 measure é cada fator. Podemos notar que para cada dataset, temos diferentes resultados, o que é importante para um não representa o mesmo para outro.

Olhando cada fator, podemos notar que o número de camadas ocultas apresenta em média 8% de variação, o número de neurônios em cada camada apresenta 41% de variação. Já para a regularização a média foi de 0,08% e para a taxa de aprendizado foi de 25%. Em geral, três fatores sozinhos (A, B e D) apresentam uma importância no desempenho. Para o fator da regularização (C), notamos que ele não implica no resultado da F1. Algumas outros outros resultados mostram que a combinação AB para o dataset Breast e Wine é importante, o que não reflete nos outros datasets.

Para o dataset Pima podemos notar que o número de neurônios nas camadas (B) sozinho não representa uma variação no resultado, mas combinado com a taxa de aprendizado (BD) ele representa 25,19% e combinado com o número de camadas e com a taxa de aprendizado (ABD) ele representa 3,21%. Outra combinação importante para este dataset é a junção de número de camadas com a taxa de aprendizado (AD) que apresenta uma variação de 12,35%. Para o dataset Breast, uma combinação que tem apresenta uma variação no desempenho da F1 é a junção dos número de

camadas com o número de neurônios (AB) que chega a 11,28%. Outras combinações para este dataset que apresentam variações são AD, BD e ABD.

Para o dataset Wine, combinações como AB, AD e ABD apresentam uma variação no desempenho da F1. No dataset Ionosphere podemos perceber que para A, este apresenta a menor variação (4,23%). Como para o Pima e Breast Cancer Wisconsin, o Ionosphere apresenta uma variação para a combinação BD e ABD. Combinações com valores 0 ou próximos a ele, não apresenta diferenças significativas nos resultados neste caso as combinações C, AC, BC, CD, ABC, ACD, BCD e ABCD, todas que contém o fator C, já que ele não apresenta uma variação. Sendo assim, concluímos que para estes datasets, os fatores que mais impactam no desempenho são o número de camadas ocultas (A), o número de neurônios nestas camada (B) e a taxa de aprendizagem (D).

	A	**B**	**C**	**D**	**AB**	**AC**	**AD**	**BC**
Pima	10,22%	0,05%	0,10%	45,44%	0,85%	0,32%	12,35%	0,63%
Breast	11,18%	59,71%	0,05%	7,37%	11,28%	0,02%	3,59%	0,05%
Wine	8,28%	69,01%	0,02%	13,05%	4,97%	0,02%	1,18%	0,01%
Ionosphere	4,23%	36,58%	0,15%	36,69%	0,24%	0,00%	0,00%	0,62%

	BD	**CD**	**ABC**	**ABD**	**ACD**	**BCD**	**ABCD**
Pima	25,19%	0,34%	1,10%	3,21%	0,00%	0,19%	0,01%
Breast	2,77%	0,00%	0,03%	3,94%	0,00%	0,00%	0,00%
Wine	0,68%	0,03%	0,03%	2,49%	0,04%	0,13%	0,05%
Ionosphere	14,29%	0,01%	0,01%	6,90%	0,17%	0,05%	0,05%

Tabela 1. Tabela de saída 2kn

network[1]	**lambda**	**alpha**	**F1 média**	**F1(σ)**	**tempo médio**	**tempo(σ)**
4+2	0	0,01	0,7885	0	1141,79	130,27
4+2	0,001	0,01	0,7885	0	1298,91	204,89
4+2	0,1	0,01	0,7885	0	1337,55	165,83
3+2	0,1	0,01	0,7903	0,006	1175,62	85,09
4+4	0,1	0,01	0,7910	0,005	1271,70	103,83
4+2	0,1	0,1	0,7944	0,009	281,93	106,49
4+2	0	0,1	0,7983	0,010	328,96	129,31

2+2	0,1	0,01	0,7999	0,012	1134,51	73,19
4+2	0,001	0,1	0,7999	0,013	355,34	163,49
3+4	0,1	0,01	0,8042	0,011	1324,52	100,46
3+4	0,001	0,01	0,8043	0,018	1286,89	278,91
3+4	0	0,01	0,8071	0,013	1375,89	163,61
1+2	0,1	0,01	0,8273	0,023	962,66	77,47
1+2	0,001	0,01	0,8275	0,015	955,98	54,14
2+4	0	0,01	0,8282	0,013	1335,67	131,03
2+4	0,001	0,01	0,8309	0,012	1297,75	47,29
2+4	0,1	0,01	0,8368	0,021	1336,82	152,80
1+2	0	0,01	0,8382	0,020	825,13	68,02
4+8	0,1	0,01	0,8451	0,030	1833,63	338,90
3+2	0,1	0,1	0,8528	0,026	817,35	219,47
3+8	0,1	0,01	0,8686	0,027	1818,03	166,14
3+8	0,001	0,01	0,8731	0,026	1806,80	179,32
4+4	0,1	0,1	0,8784	0,040	1241,65	472,12
1+4	0,1	0,01	0,8882	0,018	1066,15	17,43
3+8	0	0,01	0,8925	0,021	1887,86	155,54
2+8	0	0,01	0,8972	0,025	1498,67	50,08
2+8	0,001	0,01	0,9018	0,029	1499,02	55,49
2+8	0,1	0,01	0,9049	0,020	1535,33	54,96
2+2	0,1	0,1	0,9181	0,031	1103,04	215,54
1+8	0,1	0,01	0,9279	0,011	1080,23	10,85
3+4	0	0,1	0,9281	0,020	1480,04	216,17
3+4	0,001	0,1	0,9304	0,024	1518,12	224,64
3+4	0,1	0,1	0,9356	0,014	1529,97	129,83
1+2	0,1	0,1	0,9559	0,016	946,62	65,98
1+16	0,001	0,01	0,9564	0,007	1080,02	39,62
4+16	0	0,01	0,9571	0,008	2189,95	54,80
4+16	0,1	0,01	0,9572	0,010	2415,88	160,18

1+16	0,1	0,01	0,9579	0,006	1113,86	21,41
1+16	0	0,01	0,9587	0,008	937,39	12,62
2+16	0,1	0,01	0,9590	0,004	1617,89	22,99
3+16	0,1	0,01	0,9592	0,007	2072,57	98,90
4+16	0,001	0,01	0,9614	0,009	2368,92	196,81
4+8	0,1	0,1	0,9634	0,015	1804,32	215,98
2+4	0,001	0,1	0,9665	0,010	1390,16	81,83
2+4	0	0,1	0,9668	0,016	1398,17	155,30
1+2	0	0,1	0,9683	0,009	1017,75	58,77
1+2	0,001	0,1	0,9706	0,009	997,73	35,73
2+4	0,1	0,1	0,9716	0,013	1397,08	64,53
1+16	0	0,1	0,9727	0,004	637,17	27,54
1+16	0,1	0,1	0,9740	0,003	628,38	20,54
2+8	0,1	0,1	0,9743	0,002	1137,41	75,47
3+8	0	0,1	0,9744	0,006	1525,37	122,08
4+16	0,1	0,1	0,9747	0,003	1401,41	75,06
1+8	0,1	0,1	0,9747	0,003	778,33	32,78
1+4	0,1	0,1	0,9748	0,002	947,02	52,97
3+8	0,1	0,1	0,9749	0,003	1501,77	96,29
1+16	0,001	0,1	0,9749	0,002	**623,21**	30,02
3+8	0,001	0,1	0,9751	0,002	1469,02	81,03
4+16	0	0,1	0,9754	0,003	1333,78	119,41
4+16	0,001	0,1	0,9758	0,003	1260,17	97,18
3+16	0,1	0,1	0,9759	0,004	1130,36	50,97
2+8	0,001	0,1	0,9760	0,003	1126,91	104,76
2+8	0	0,1	0,9763	0,002	1132,80	75,57
2+16	0,1	0,1	**0,9764**	0,003	830,17	55,15

1. Coluna Network representa Camadas+Neurônios em cada camada.

Tabela 2. Resultados dataset breast cancer wisconsin

Com base nos resultados obtidos, nota-se que para o conjunto de dados Breast Cancer, o modelo de rede com 2 camadas, 16 neurônios, taxa de aprendizado 0,1 e taxa de regularização 0,1

foi o que apresentou melhor performance, apresentando também um bom tempo de execução. Mantendo-se as mesmas configurações das taxas de aprendizado, taxa de regularização e adicionando uma camada (3 camadas e 16 neurônios), houve pouca diferença na performance, porém houve redução no tempo de execução, porém quando adicionamos uma nova camada sem alterar as mesmas configurações nota-se que o tempo de execução aumenta. A rede que levou o menor tempo de execução foi a rede com 1 camada e 16 neurônios, taxa de regularização 0,001 e taxa de aprendizagem 0,1, não sofrendo grandes perdas com relação ao desempenho.

network[1]	**lambda**	**alpha**	**F1 média**	**F1(σ)**	**tempo médio**	**tempo(σ)**
3+2	0,1	0,1	0,7507	0,041	72,48	4,07
3+16	0,1	0,01	0,7532	0,056	368,12	120,31
3+4	0,1	0,01	0,7586	0,038	330,14	22,24
4+2	0	0,1	0,7586	0,038	87,25	10,07
2+8	0,001	0,01	0,7595	0,036	243,02	28,49
1+2	0	0,01	0,7618	0,052	195,79	24,83
1+4	0,1	0,01	0,7629	0,037	243,59	26,25
2+4	0	0,01	0,7660	0,034	266,33	42,24
4+2	0	0,01	0,7664	0,033	422,60	52,45
3+8	0,1	0,01	0,7669	0,033	249,98	19,42
4+16	0,1	0,01	0,7679	0,034	269,54	75,57
2+2	0,1	0,01	0,7743	0,025	306,43	51,60
3+2	0,1	0,01	0,7743	0,025	424,46	56,49
3+4	0	0,01	0,7743	0,025	350,24	37,29
3+4	0,001	0,01	0,7743	0,025	359,87	33,59
3+4	0,1	0,1	0,7743	0,025	**58,92**	4,79
3+8	0	0,01	0,7743	0,025	259,49	34,30
3+8	0,001	0,01	0,7743	0,025	266,07	39,42
4+2	0,001	0,01	0,7743	0,025	467,08	57,85
4+2	0,001	0,1	0,7743	0,025	86,31	10,89
4+2	0,1	0,01	0,7743	0,025	499,13	59,88
4+2	0,1	0,1	0,7743	0,025	86,29	8,00
4+4	0,1	0,01	0,7743	0,025	415,90	64,71

4+4	0,1	0,1	0,7743	0,025	70,53	7,81
2+4	0,1	0,01	0,7744	0,025	274,08	38,32
4+16	0	0,01	0,7757	0,026	277,59	98,66
2+8	0	0,01	0,7765	0,025	259,42	47,15
1+2	0,1	0,01	0,7772	0,026	245,74	25,05
2+2	0,1	0,1	0,7791	0,022	93,93	49,27
4+8	0,1	0,01	0,7821	0,000	328,31	42,11
2+4	0,001	0,01	0,7823	0,000	253,31	23,14
4+16	0,001	0,01	0,7833	0,001	242,82	63,01
2+8	0,1	0,01	0,7842	0,002	236,80	29,90
1+2	0,001	0,01	0,7846	0,002	244,96	23,30
3+4	0	0,1	0,7861	0,007	99,12	70,03
3+4	0,001	0,1	0,7862	0,012	99,84	93,39
2+16	0,1	0,01	0,7870	0,036	614,08	99,03
1+8	0,1	0,01	0,7892	0,026	337,07	56,27
4+8	0,1	0,1	0,7905	0,011	126,75	91,19
2+4	0	0,1	0,8057	0,033	337,80	155,84
1+16	0	0,01	0,8170	0,023	480,14	9,25
1+2	0	0,1	0,8181	0,055	376,91	73,77
1+16	0,001	0,01	0,8184	0,010	543,62	27,56
1+16	0,1	0,01	0,8226	0,014	530,22	22,77
2+4	0,1	0,1	0,8248	0,021	379,71	119,94
2+4	0,001	0,1	0,8254	0,019	374,94	130,72
1+2	0,1	0,1	0,8327	0,022	401,44	68,27
3+8	0	0,1	0,8332	0,014	468,61	90,54
1+2	0,001	0,1	0,8355	0,037	333,46	61,76
3+8	0,001	0,1	0,8385	0,026	463,85	205,32
3+8	0,1	0,1	0,8392	0,021	506,04	153,14
1+4	0,1	0,1	0,8754	0,029	543,11	24,73

1+8	0,1	0,1	0,8847	0,022	560,27	9,85
2+8	0	0,1	0,8876	0,031	757,38	52,39
2+8	0,1	0,1	0,8907	0,030	770,32	32,37
1+16	0,1	0,1	0,8939	0,034	572,40	9,08
2+8	0,001	0,1	0,8942	0,017	733,93	83,26
1+16	0,001	0,1	0,8958	0,042	524,63	32,05
1+16	0	0,1	0,9099	0,027	557,41	27,76
4+16	0,001	0,1	0,9116	0,036	1106,60	84,36
2+16	0,1	0,1	0,9138	0,032	828,57	13,68
4+16	0,1	0,1	0,9139	0,015	1172,97	135,03
4+16	0	0,1	0,9143	0,013	1121,73	87,72
3+16	0,1	0,1	**0,9231**	0,014	1083,34	29,73

1. Coluna Network representa Camadas+Neurônios em cada camada.

Tabela 3. Resultados dataset Ionosphere

Com base nos resultados obtidos, para o conjunto de dados Ionosphere a configuração de rede que apresentou melhor desempenho foi utilizando-se 3 camadas e 16 neurônios, com taxa de regularização 0,1 e taxa de aprendizagem 0,1. Mantendo-se a taxa de regularização e aprendizagem e reduzindo o numero de camadas para 2 camadas com 16 neurônios, houve decréscimo no tempo de execução, sem grandes alterações no desempenho. A rede que apresentou menor tempo de execução foi utilizando 3 camadas com 4 neurônios, taxa de regularização 0,1 e taxa de aprendizagem 0,1, porém houve grande perda de performance.

network[1]	**lambda**	**alpha**	**F1 média**	**F1(σ)**	**tempo médio**	**tempo(σ)**
1+16	0	0,01	0,7487	0,022	883,54	197,48
1+16	0,1	0,01	0,7544	0,020	824,73	103,67
1+16	0,001	0,01	0,7547	0,020	771,60	99,59
1+8	0,1	0,01	0,7651	0,011	487,10	51,68
1+2	0,1	0,01	0,7739	0,014	681,65	89,10
1+4	0,1	0,01	0,7745	0,012	561,48	26,81
2+16	0,1	0,01	0,7753	0,011	1095,95	176,38
1+2	0	0,01	0,7780	0,009	626,43	72,72

4+16	0,1	0,01	0,7796	0,009	1141,18	382,02
2+8	0,1	0,01	0,7805	0,011	601,85	67,40
2+8	0	0,01	0,7805	0,006	558,86	49,20
1+2	0,001	0,01	0,7819	0,013	690,31	72,78
2+8	0,001	0,01	0,7820	0,005	604,96	43,76
3+16	0,1	0,01	0,7834	0,012	1163,81	207,80
4+16	0	0,01	0,7848	0,011	1270,58	256,30
1+2	0,1	0,1	0,7860	0,006	377,18	132,70
2+4	0,1	0,01	0,7864	0,003	826,44	91,54
2+4	0,001	0,01	0,7866	0,003	806,21	128,12
2+4	0	0,1	0,7870	0,005	550,19	187,84
2+4	0	0,01	0,7872	0,003	797,49	128,80
3+8	0,001	0,01	0,7872	0,004	697,32	69,82
3+8	0	0,01	0,7879	0,002	695,59	132,59
4+16	0,001	0,01	0,7880	0,006	1112,14	241,59
3+4	0,001	0,1	0,7882	0,003	300,13	141,22
1+2	0	0,1	0,7882	0,004	327,43	119,01
2+2	0,1	0,01	0,7883	0,001	937,13	137,11
3+4	0,1	0,1	0,7883	0,004	299,76	147,21
3+8	0,1	0,01	0,7883	0,003	684,20	87,50
3+4	0,1	0,01	0,7885	0,001	926,93	114,31
3+4	0	0,01	0,7886	0,001	951,66	116,23
3+4	0,001	0,01	0,7888	0,000	1057,14	136,19
3+2	0,1	0,01	0,7889	0,000	1259,72	212,16
4+2	0	0,01	0,7889	0,000	1521,72	198,39
4+2	0	0,1	0,7889	0,000	**224,90**	20,26
4+2	0,001	0,01	0,7889	0,000	1473,60	111,21
4+2	0,001	0,1	0,7889	0,000	233,38	21,24
4+2	0,1	0,01	0,7889	0,000	1597,97	221,84

4+2	0,1	0,1	0,7889	0,000	254,02	25,66
4+4	0,1	0,01	0,7889	0,000	1179,01	192,71
2+2	0,1	0,1	0,7890	0,001	199,03	90,37
4+8	0,1	0,01	0,7890	0,001	846,04	110,18
2+4	0,001	0,1	0,7890	0,007	387,38	133,01
3+4	0	0,1	0,7891	0,004	320,18	228,89
4+4	0,1	0,1	0,7891	0,005	266,15	130,35
3+2	0,1	0,1	0,7895	0,002	245,99	93,21
3+8	0	0,1	0,7898	0,016	998,01	307,09
2+8	0,1	0,1	0,7900	0,016	1155,78	238,77
2+8	0,001	0,1	0,7902	0,015	1221,08	272,86
1+2	0,001	0,1	0,7908	0,006	390,41	107,77
2+4	0,1	0,1	0,7915	0,006	652,12	249,59
4+8	0,1	0,1	0,7926	0,005	917,95	255,56
1+4	0,1	0,1	0,7929	0,010	708,26	143,25
3+8	0,001	0,1	0,7929	0,009	1080,84	426,64
3+8	0,1	0,1	0,7939	0,004	1220,79	363,09
2+8	0	0,1	0,7972	0,009	1203,17	156,37
1+8	0,1	0,1	0,8000	0,010	1113,25	73,37
1+16	0	0,1	0,8010	0,010	1137,37	65,95
4+16	0	0,1	0,8022	0,012	2077,71	132,34
3+16	0,1	0,1	0,8025	0,012	1793,93	246,55
4+16	0,1	0,1	0,8040	0,015	2003,63	300,56
4+16	0,001	0,1	0,8060	0,011	2068,65	196,77
2+16	0,1	0,1	0,8076	0,013	1526,12	107,21
1+16	0,001	0,1	0,8084	0,009	1079,16	82,56
1+16	0,1	0,1	**0,8130**	0,007	1176,83	42,30

1. Coluna Network representa Camadas+Neurônios em cada camada.

Tabela 4. Resultados dataset Pima

Para o conjunto de dados Pima, o melhor desempenho foi obtido pela rede com uma camada

e 16 neurônios, taxa de regularização 0,1 e taxa de aprendizagem 0,1. Ao alterar apenas o número de camadas, de uma camada com 16 neurônios para 2 camadas com 16 neurônios, e mantendo-se as demais configurações, houve perda de performance e acréscimo no tempo de execução. O menor tempo de execução foi obtido utilizando rede com 2 camadas e 4 neurônios, porém houve grande redução na performance.

network[1]	lambda	alpha	F1 média	F1(σ)	tempo médio	tempo(σ)
4+2	0	0,01	0,2826	0,009	358,43	23,67
4+2	0,001	0,01	0,2830	0,029	360,21	41,30
3+2	0,1	0,01	0,2894	0,015	325,19	18,09
4+2	0,1	0,01	0,2914	0,017	357,40	37,46
4+2	0,1	0,1	0,2930	0,000	**59,97**	5,56
4+2	0	0,1	0,2935	0,002	71,00	22,69
4+2	0,001	0,1	0,2936	0,002	64,16	18,94
3+2	0,1	0,1	0,3110	0,022	93,99	40,21
4+4	0,1	0,01	0,3186	0,038	301,43	32,60
3+4	0,001	0,01	0,3259	0,036	250,78	30,08
3+4	0,1	0,01	0,3298	0,036	269,00	30,22
2+2	0,1	0,01	0,3364	0,025	247,30	20,02
3+4	0	0,01	0,3383	0,061	258,47	27,43
4+4	0,1	0,1	0,3604	0,046	183,59	70,78
2+4	0,1	0,01	0,3714	0,058	260,48	27,20
4+8	0,1	0,01	0,3817	0,062	329,66	62,42
1+2	0	0,01	0,4012	0,056	230,20	11,80
1+2	0,1	0,01	0,4217	0,042	230,89	21,80
2+4	0,001	0,01	0,4335	0,063	269,20	35,10
2+4	0	0,01	0,4568	0,069	265,94	40,23
3+8	0	0,01	0,4603	0,056	364,25	30,30
1+2	0,001	0,01	0,4707	0,091	232,39	26,50
2+2	0,1	0,1	0,4754	0,075	218,16	50,79
3+8	0,001	0,01	0,4786	0,041	367,89	43,93

3+8	0,1	0,01	0,4906	0,105	388,34	51,21
1+4	0,1	0,01	0,5330	0,064	258,75	11,34
3+4	0,001	0,1	0,5463	0,127	306,88	108,41
3+4	0,1	0,1	0,5498	0,074	341,86	57,65
3+4	0	0,1	0,5733	0,088	320,29	68,14
2+8	0	0,01	0,5769	0,089	376,98	26,26
2+8	0,1	0,01	0,5988	0,069	380,93	16,39
2+8	0,001	0,01	0,6018	0,075	370,42	24,89
1+8	0,1	0,01	0,6465	0,059	278,02	4,37
1+2	0,1	0,1	0,6508	0,098	236,95	25,85
4+16	0,1	0,01	0,6942	0,036	623,29	39,62
4+16	0,001	0,01	0,7070	0,066	613,21	48,71
4+16	0	0,01	0,7088	0,049	614,12	48,96
1+2	0	0,1	0,7278	0,050	246,50	21,76
1+2	0,001	0,1	0,7394	0,077	230,32	20,61
3+16	0,1	0,01	0,7420	0,064	531,36	15,96
1+16	0,1	0,01	0,7532	0,039	278,90	3,48
1+16	0,001	0,01	0,7632	0,048	281,09	5,49
1+16	0	0,01	0,7659	0,058	278,95	4,65
2+4	0,001	0,1	0,7758	0,043	350,41	25,06
4+8	0,1	0,1	0,7805	0,108	535,56	93,27
2+16	0,1	0,01	0,7819	0,055	412,65	8,95
2+4	0,1	0,1	0,7896	0,058	368,39	22,51
2+4	0	0,1	0,7958	0,077	370,33	26,57
3+8	0,1	0,1	0,8912	0,039	499,09	26,30
3+8	0	0,1	0,8946	0,047	501,28	31,86
1+4	0,1	0,1	0,9080	0,043	270,99	7,85
3+8	0,001	0,1	0,9223	0,027	517,91	20,61
1+8	0,1	0,1	0,9377	0,011	272,62	6,84

2+8	0,001	0,1	0,9383	0,014	386,22	15,25
2+8	0	0,1	0,9431	0,018	394,41	9,78
2+8	0,1	0,1	0,9481	0,016	391,55	11,26
4+16	0	0,1	0,9502	0,013	618,08	41,31
1+16	0,001	0,1	0,9511	0,016	256,52	10,48
2+16	0,1	0,1	0,9535	0,013	404,67	10,87
4+16	0,001	0,1	0,9537	0,018	587,84	28,62
4+16	0,1	0,1	0,9552	0,016	626,60	39,80
1+16	0	0,1	0,9565	0,014	283,03	8,92
3+16	0,1	0,1	0,9650	0,013	513,28	28,55
1+16	0,1	0,1	**0,9651**	0,011	280,72	7,71

1. Coluna Network representa Camadas+Neurônios em cada camada.

Tabela 5. Resultados dataset Wine

Para o conjunto de dados Wine, a rede que apresentou melhor desempenho possui uma camada com 16 neurônios, taxa de regularização 0,1 e taxa de aprendizagem 0,1. Mantendo-se as configurações das taxas de regularização e aprendizagem e alterando-se para 3 camadas com 16 neurônios, não houve alteração significativa na performance, porém o tempo de execução aumentou consideravelmente. A configuração de rede que apresentou o menor tempo de execução foi a rede com 4 camadas com 2 neurônios, taxa de regularização 0,1 e taxa de aprendizagem 0,1, porém a performance foi prejudicada drasticamente.

Avaliação da performance da rede para os conjuntos de dados:

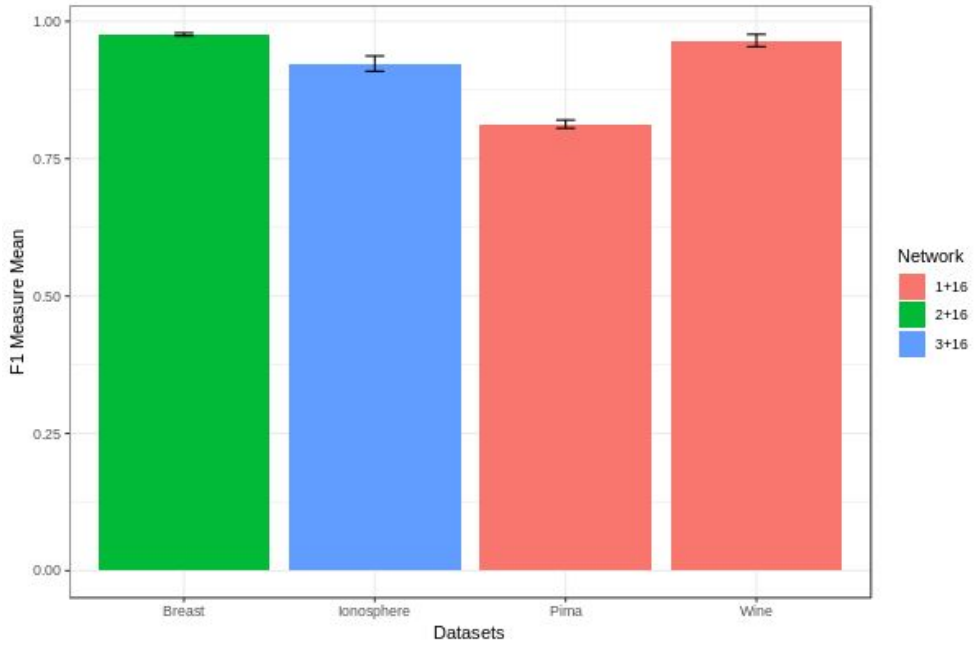

Figura 7. Melhores Configurações para cada dataset testado

As melhores configurações para todos os datasets testados utilizaram o lambda e o alpha iguais a 0.1. A melhor configuração da rede neural para o dataset breast cancer foi 2 camadas ocultas e 16 neurônios. Já para o dataset wine e Pima a melhor configuração encontrada foi 1 camada oculta e 16 neurônios. A melhor configuração encontrada para o dataset Ionosphere foi 3 camadas ocultas e 16 neurônios.

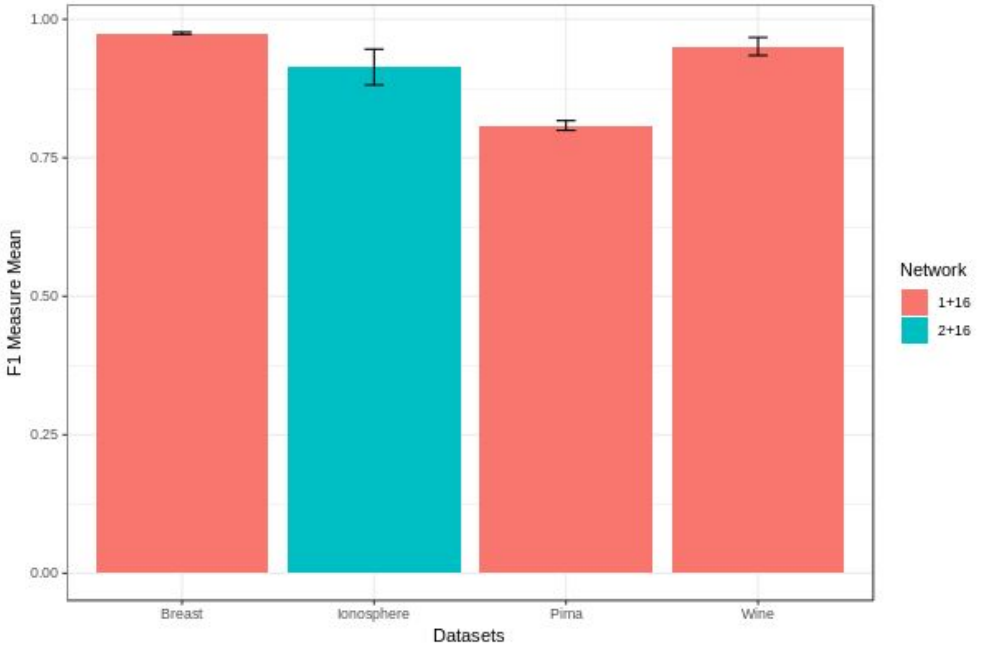

Figura 8. Performance da rede com menor tempo de execução e F1 maior que 0.96

Melhor resultado se pegar o menor tempo de execução, filtrando para o Breast uma f1 maior que 0.96 para o Pima maior que 0.8 para o Wine maior que 0.95 e para o Ionosphere de 0.91

5.3 Curvas de Aprendizado

Utilizando a melhor configuração da rede neural para cada dataset foi gerado uma curva de aprendizado mostrando a função de custo J para cada dataset. As condições de parada para o treinamento das redes foram: o número máximo de iterações igual a 1000 e se a diferença entre o J atual e o anterior for igual a 0.0001.

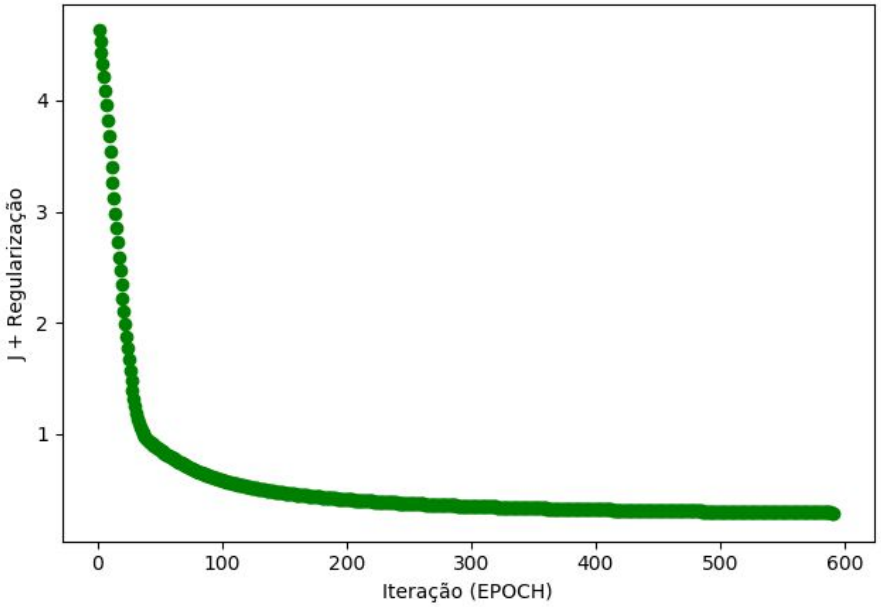

Figura 9. Variação do J com o número de iterações para a melhor configuração da rede neural para o dataset Breast Cancer Wisconsin

Na figura acima temos a variação de J para cada iteração para o dataset Breast Cancer Wisconsin percebemos que próximo de 100 iterações o valor de J decai para abaixo de 1 começando a estabilizar o seu valor e sua execução termina com o número máximo de iterações definido 1000.

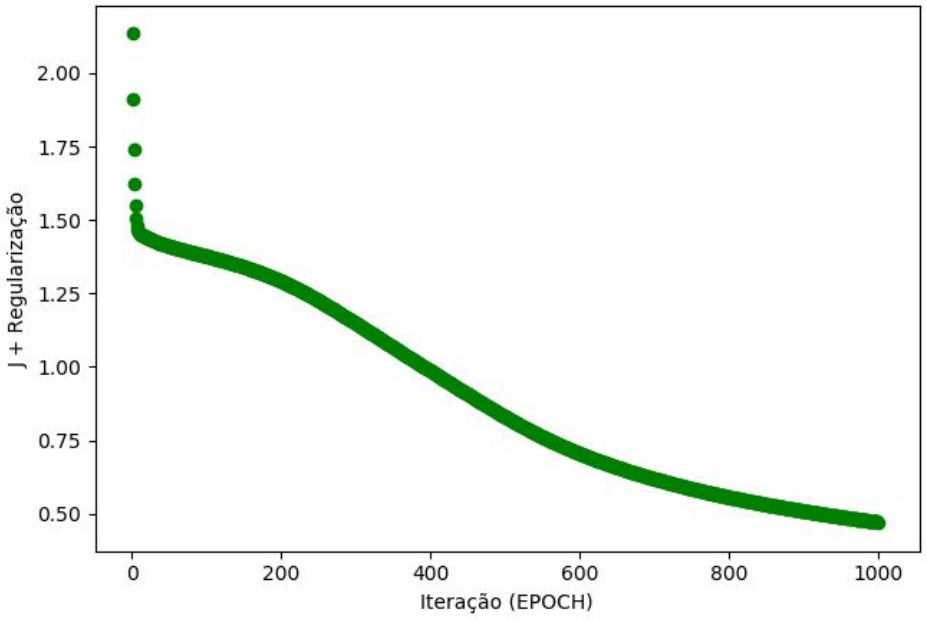

Figura 10. Variação do J com o número de iterações para a melhor configuração da rede neural para o dataset Ionosphere

Na figura acima temos a variação de J para cada iteração para o dataset Ionosphere percebemos que próximo de 200 iterações o valor de J volta a crescer depois volta a cair novamente depois das 300 iterações. Este efeito pode representar um mínimo local. A rede neural começa a

estabilizar perto de 1000 iterações e sua execução termina com o número máximo de iterações que era igual a 1000.

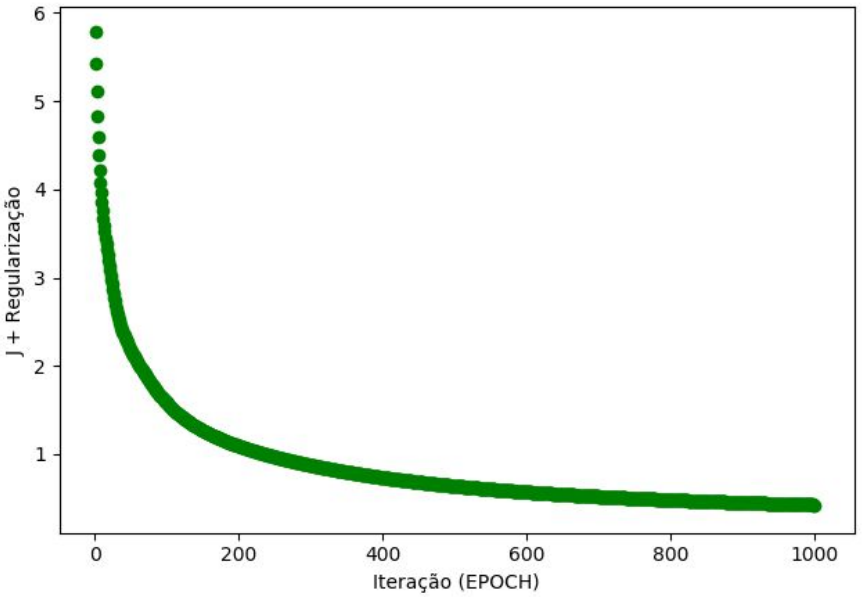

Figura 11. Variação do J com o número de iterações para a melhor configuração da rede neural para o dataset Wine

Na figura acima temos a variação de J para cada iteração para o dataset Wine percebemos que próximo de 400 iterações o valor de J decai para abaixo de 1 começando a estabilizar o seu valor e sua execução termina com o número máximo de iterações definido 1000

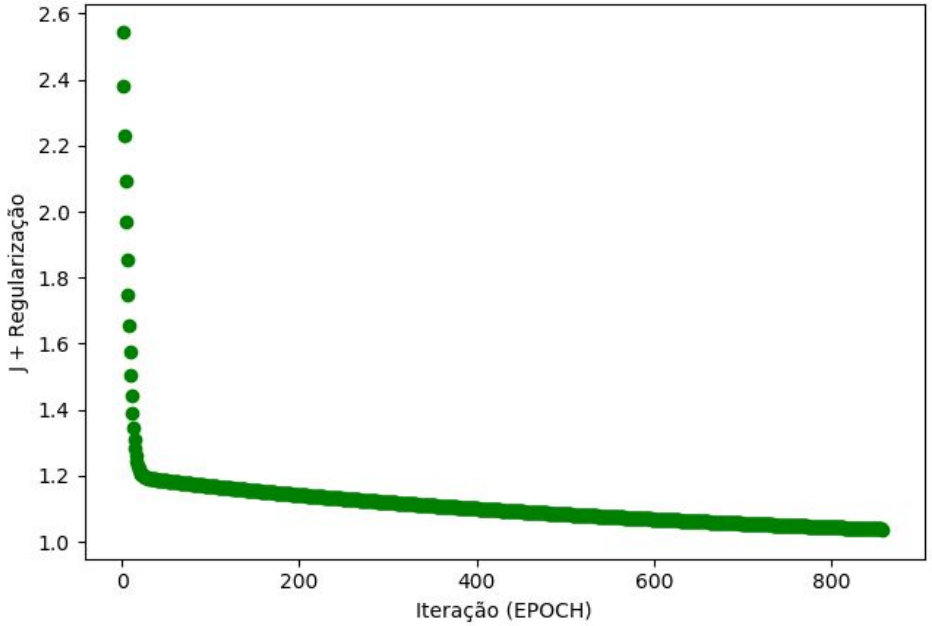

Figura 12. Variação do J com o número de iterações para a melhor configuração da rede neural para o dataset Pima

Na figura acima temos a variação de J para cada iteração para o dataset pima percebemos que após alguma iterações o valor de J decai para abaixo de 1 começando a estabilizar o seu valor e

sua execução termina com um pouco mais de 900 iterações.

6. Conclusão

Este trabalho teve como objetivo realizar o treinamento de redes neurais feedforward com diferentes configurações, utilizando o algoritmo de backpropagation em busca dos pesos que apresentam o melhor desempenho. Foram testadas redes neurais com diferentes números de camadas e neurônios em cada camada, taxas de regularização e taxas de aprendizagem. Foram avaliados a performance da rede utilizando o método de validação cruzada. Além disso, foi avaliado o tempo necessário para a convergência da rede.

Os melhores resultados foram obtidos com redes com 16 neurônios e o número de camadas variando de 1 até 3, taxa de aprendizagem 0,1 e taxa de regularização 0,1. O conjunto de dados Breast Cancer apresentou melhor desempenho com rede de 2 camadas e 16 neurônios, e a partir de 100 iterações o custo decai para valores abaixo de 1, e continua melhorando até atingir o número máximo de 1000 iterações. O conjunto de dados Ionosphere apresentou melhor desempenho com rede de 3 camadas e 16 neurônios, onde foi possível notar que próximo de 200 iterações o custo voltou a crescer, voltando a diminuir a partir de 300 iterações, continuando a reduzir até atingir o máximo estabelecido de 1000 iterações. O conjunto de dados Pima apresentou melhores resultados com rede de uma camada e 16 neurônios, onde o valor do custo apresentou bons resultados após poucas iterações, e encontra os melhores pesos antes de atingir o número máximo de iterações pré-estabelecido. O conjunto de dados Wine apresentou melhor desempenho para a rede com uma camada e 16 neurônios, onde percebe-se que o custo atinge valor menor que 1 após 400 iterações, continuando a decrescer até atingir o máximo de iterações pré-estabelecido de 1000 iterações.

7. Referências

FACELI, Katti et al. Inteligência Artificial: Uma Abordagem de Aprendizado de Máquina. Rio de Janeiro: LTC, 2011

RUSSEL, Stuart J.; NORVIG, Peter. Inteligência artificial. Rio de Janeiro: Elsevier Campus, 2004.

GRUS, Joel. Data Science do Zero. Rio de Janeiro: Alta Books, 2016.

DEEP LEARNING BOOK. Disponível em <http://deeplearningbook.com.br/capitulos/> . Acesso em: 30 nov 2018.

JAIN, Raj. The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling. John Wiley & Sons, 1990.