

Classificação de tumores utilizando Redes Neurais Artificiais

Cesar Augusto de Carvalho

Departamento de Engenharia Elétrica

Universidade de Brasília

Matrícula: 15/0121814

Gabriel da Silva Lima

Departamento de Engenharia Elétrica

Universidade de Brasília

Matrícula : 17/0170527

Renan Raeder Ribeiro

Departamento de Engenharia Elétrica

Universidade de Brasília

Matrícula : 14/0056955

Resumo—Este documento tem por objetivo descrever a funcionalidade do trabalho desenvolvido na disciplina de Inteligência Artificial.

Index Terms—*Machine learning, artificial intelligence, diagnostic*

I. INTRODUÇÃO

Este projeto consiste em aplicar conhecimentos obtidos na disciplina de Inteligência Artificial para a implementação de algoritmos de *machine learning*, visando a obtenção de análises práticas sobre problemas reais tais como o diagnóstico de câncer de mama usando *deep learning* na análise de milhares de imagens de células cancerígenas e reconhecimento de padrões; na área de recursos humanos, avaliando os candidatos e listando os que apresentam maior chance de serem bem sucedidos de acordo com o perfil da equipe, ou na área financeira, em que a IA ajuda a tomar decisões mais acertadas e com rapidez, mesmo com muitas variáveis, pois é capaz de se beneficiar da capacidade de processamento e análise de um grande volume de dados. Percebe-se, portanto, a polivalência da inteligência artificial sobre os mais variados campos de atuação, seja em problemas práticos ou análises puramente acadêmicas.

II. CONTEXTUALIZAÇÃO DO PROBLEMA

A. Câncer de mama

De acordo com o INCA (Instituto Nacional de Câncer), câncer pode ser designado como um conjunto de mais de 100 doenças que apresentam algum tipo de característica de crescimento desordenado de células, que podem invadir tecidos e órgãos. Os fatores para formação de um câncer podem ser de origem interna ou externa, sendo que os fatores internos em geral são associados a fatores genéticos pré-determinados. Também de acordo com o INCA, o câncer de mama é o que mais acomete mulheres no Brasil e no mundo depois do câncer de pele não melanoma.

Apesar de ser esmagadoramente preponderante em pessoas do gênero feminino, o câncer de mama também ocorre em pessoas do gênero masculino, porém corresponde a um percentual extremamente baixo. O fator de maior relação com o câncer de mama é a idade. A faixa-etária que o câncer atinge um maior número de pessoas é a dos 50-60 anos, tendo boa diminuição da taxa de crescimento da incidência em idades mais elevadas.

Porém, existem outros elementos de risco que podem atuar para a formação de um câncer de mama, entre eles a vida reprodutiva da mulher, o histórico familiar de câncer de mama e a alta densidade do tecido mamário.

Dentro do espaço de possibilidades de diferentes tipos de câncer, o de mama é tratado como um de boa chance de prognóstico, baseados em todos os diagnósticos presentes na literatura da área. Porém, a ressalva maior fica em torno de ainda existirem altos índices de mortalidade, o que provavelmente se explica pela ineficiência de uma detecção em estágios iniciais da doença. O exame clínico de mama é ainda hoje a forma mais eficiente e paupável tecnologicamente para um grande número de pessoas. A mamografia também se apresenta como uma boa ferramenta para detecção (diagnóstico), contudo, depende em demasia da atuação física do radiologista.

B. Objetivos

A ideia de uma enorme base de dados com diferentes imagens de tumores dos mais diversos formatos permite uma abordagem por *Machine Learning*. Os conceitos da área de inteligência artificial podem ser de grande valia em caso de busca por um processo de discriminação de padrões de benignidade e malignidade de massas nas imagens obtidas em mamografia.

O possível desenvolvimento de códigos robustos para atuação na detecção de tumores viabiliza o fortalecimento dos diagnósticos médicos, diminuindo a dependência de um olhar clínico humano, possibilitando uma análise científica mais fundamentada e menos sujeita a erros.

Neste trabalho em específico o objetivo é classificar em tumor maligno ou tumor benigno um conjunto de dados retirado de imagens e pré-processado de acordo com algumas características selecionadas (conforme será explicado mais a frente).

III. APRESENTAÇÃO TEÓRICA

A. Perceptron multicamadas

O perceptron multicamadas pode ser entendido como uma otimização do modelo de rede neural em que a camada de entrada (não realiza nenhum tipo de processamento) se conecta a camada de saída, conhecido como Perceptron. Este modelo com apenas um neurônio na camada de saída mostra-se ineficiente dada sua não capacidade de resolução de

problemas não-linearmente separáveis. Decide-se, então, por uma interconexão das entradas à camada de saída por pelo menos uma camada de neurônios intermediária, tipicamente denominada de camada escondida. O resultado é uma ampla gama de possíveis problemas solucionados por essas redes.

Existem variadas formas de classificação de uma rede neural artificial. Uma das mais importantes formas é a classificação pelo processo ou algoritmo de aprendizado empregado para treiná-la, o qual pode ser supervisionado ou não-supervisionado.

No aprendizado supervisionado, a rede recebe uma série de padrões ou vetores de entradas com sua respectiva resposta ou saída desejada. A comparação entre a saída desejada e a saída obtida gerada pela rede é utilizada para alteração dos parâmetros internos (ou pesos sinápticos) da rede de maneira a aproximar sua resposta a saída desejada. Este procedimento é feito até que a diferença entre as saídas obtida pela rede para os diversos padrões apresentados e a resposta esperada seja menor que um determinado limiar predefinido.

O processo de treinamento ou aprendizado supervisionado de uma rede neural consiste, essencialmente, no processo de minimização do erro entre a saída obtida pela rede para um determinado padrão de entrada e a resposta desejada para aquele mesmo padrão. Para o escopo deste trabalho foi escolhido processo supervisionado mais vezes utilizado para treinamento de redes neurais MLP, o *backpropagation*.

O algoritmo de *backpropagation* consiste em duas fases: a primeira é o *forward pass*, isto é, alimentar a rede a partir da entrada, passando por toda rede, até a saída, onde são obtidas as previsões. A segunda etapa é a *backward pass* (ou *backpropagation*), que propaga o gradiente da função de perda da camada de saída e a regra da cadeia é aplicada recursivamente para trás (da camada de saída até a camada de entrada).

A etapa de *backpropagation* consistiu em um grande problema durante muito tempo, tratado como um processo limitado devido à falta de possibilidade de utilização de camadas escondidas. O "pulo do gato" foi a percepção de que se poderia utilizar a regra da cadeia no algoritmo.

Nesta etapa, deseja-se atualizar os pesos de acordo com a comparação entre a saída recebida e a saída desejada, utilizando descida de gradiente, a fim de encontrar o ponto mínimo da função:

$$\Delta w_i = -\eta \frac{\partial E}{\partial w_i}$$

Como mostrado na Figura 1, o erro das camadas anteriores é calculado por meio da utilização da chamada derivada da função de custo (também conhecida por *Loss function*).

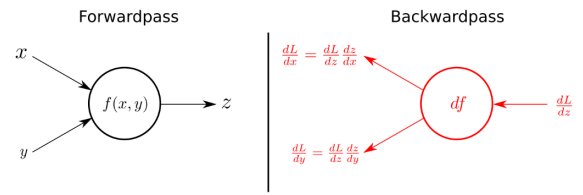


Figura 1. Resumo do funcionamento do *forward pass* e *backward pass*.

B. Função de custo

Uma função de custo comumente utilizada é a entropia cruzada:

$$C = \frac{1}{n} \sum_x [y \ln a + (1 - y) \ln 1 - a]$$

em que y é o valor desejado, e a é o valor obtido.

C. Outras técnicas utilizadas

Quanto a outros estilos de treinamento, podemos realizar o treinamento *on-line*, ou seja, atualizar os pesos a cada exemplo ou *batch*, que atualiza os pesos ao fim de cada época, quando todos os exemplos já passaram pela rede. O mais comum atualmente, no entanto, é a utilização de um *mini-batch*, ou seja, atualização dos pesos a cada número n de exemplos.

Uma função normalmente utilizada na camada de saída de redes neurais é a Regressão Logística (*softmax*), a qual trata os valores de um vetor como probabilidades, e assim atribui limites para seus possíveis valores.

Um problema característico ocorre quando a rede se especializa em determinados padrões contidos no conjunto inteiro de exemplos de treinamento, mas apresenta déficit para reconhecimento de outros padrões. Isto é conhecido como *overfitting*. Em caso de possibilidade de treinamento com uma base de dados extensa, o fenômeno de *overfitting* tende a apresentar significativa diminuição. Algumas vezes, contudo, a base de dados apresenta-se como limitada, e portanto, deve-se buscar a utilização de técnicas para redução do *overfitting* por meio de pré-processamento.

Uma técnica bastante comum utilizada com a intensão de reduzir o *overfitting* é a regularização L2, em que um termo é adicionado à função custo.

$$C = C_0 + \frac{\lambda}{2n} \sum_w w^2$$

Onde λ é conhecido como parâmetro de regularização e n é o tamanho do conjunto de treinamento.

A regularização L2 é semelhante à L1, porém o termo que é adicionado à função custo é diferente:

$$C = C_0 + \frac{\lambda}{n} \sum_w |w|$$

Outra técnica comum para redução do *overfitting* da rede é o *Dropout*, que consiste em retirar aleatoriamente alguns neurônios de uma ou mais camadas escondidas, e fazer com que a rede realize o treinamento sem eles. Em seguida,

estes que foram retirados são ativados novamente, e outros neurônios são retirados. A intensão do procedimento é tornar a rede mais robusta e menos dependente de neurônios específicos.

D. Arquiteturas de Rede

O termo arquitetura de uma rede neural refere-se à disposição dos neurônios, um em relação ao outro, seguindo as conexões sinápticas estudadas. A topologia da rede pode ser vista como as possíveis diferentes composições estruturais, com número diferente de quantidade de neurônios nas camadas de entrada, intermediária e também de saída da rede.

A camada de entrada é a responsável pelo recebimento dos dados/sinais/amostras que serão analisados, assim como a gerar adequadamente associação com os pesos de entrada. A camada intermediária ou escondida tem por finalidade extrair as informações associadas ao sistema trabalhado, sendo também responsável pela maior parte do processamento destes dados. Já a camada de saída é a responsável por coletar os dados das camadas anteriores e ativar uma resposta adequada.

Tipos de arquitetura de rede:

1) *FeedForward de Camada Simples*: Uma camada de entrada associa-se a um ou mais neurônios que serão responsáveis por gerar a resposta de saída. A Figura 2 demonstra que o fluxo de dados segue sempre em direção à camada de saída. Muito utilizado em problemas de classificação de padrões e filtragem. Exemplos: Perceptron e Adaline.

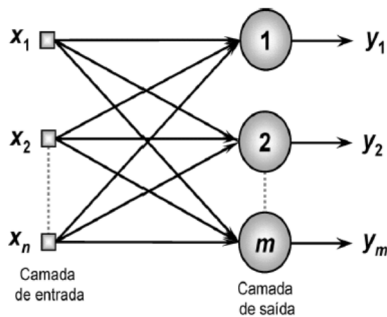


Figura 2. Arquitetura FeedForward de Camada Simples.

2) *FeedForward de Camadas Múltiplas*: É construída por uma ou mais camadas escondidas de neurônios (Figura 3). São empregadas em problemas de aproximação de funções, classificação de padrões, identificação de sistemas, otimização, robótica e controle de processos. Exemplos: Perceptron Multicamadas, Redes de Base Radial.

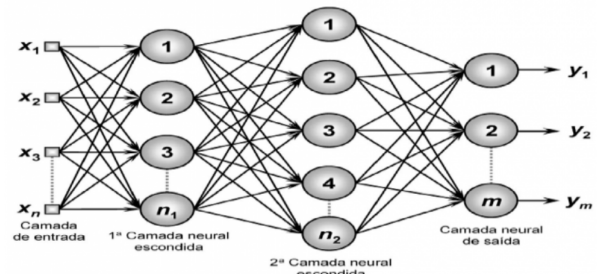


Figura 3. Arquitetura FeedForward de Camadas Múltiplas.

3) *Recorrente ou Realimentada*: As saídas da rede são realimentadas como sinais de entrada para outros neurônios, sendo assim utilizadas para o processamento de sistemas variantes no tempo (Figura 4). São empregadas em previsões de séries temporais, otimização, identificação de sistemas e controle de processos. Exemplos: Hopfield e Perceptron Multicamadas com Realimentação.

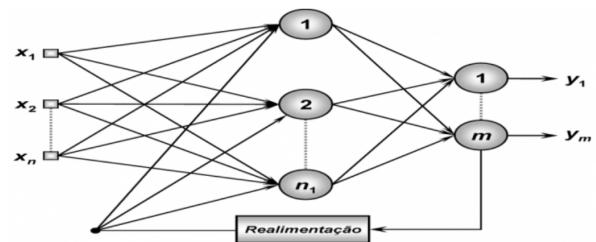


Figura 4. Arquitetura Recorrente ou Realimentada.

4) *Estrutura Reticulada*: Possuem como foco a disposição espacial dos neurônios (Figura 5) com o propósito da extração de características do sistema, isto é, sua localização espacial serve para ajuste de seus pesos e limiares. São empregadas em problemas de agrupamento, reconhecimento de padrões, otimização de sistemas, etc. Exemplos: Kohonen.

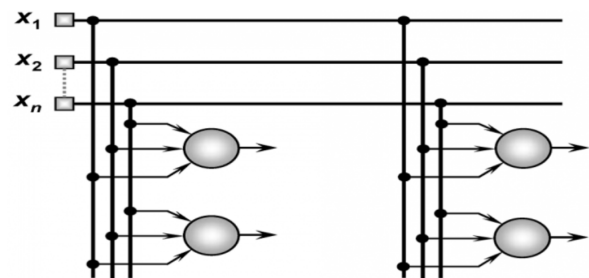


Figura 5. Arquitetura de Estrutura Reticulada.

IV. SOBRE O CONJUNTO DE DADOS

Utiliza-se o conjunto *Breast Cancer Wisconsin (Diagnostic) Data Set*, cujas características foram calculadas a partir de imagens digitalizadas por um aspirador de agulha fina (PAAF) em uma massa mamária. Estas características descrevem núcleos celulares presentes na imagem.

Dez características reais são calculadas para cada núcleo da célula:

- Raio (média das distâncias do centro para os pontos no perímetro)
- textura (desvio padrão dos valores da escala de cinza)
- perímetro
- área
- suavidade (variação local no comprimento do raio)
- compactação ($perimetro^2/area - 1.0$)
- concavidade (gravidade das porções côncavas do contorno)
- pontos côncavos (número de porções côncavas do contorno)
- simetria
- dimensão fractal ("aproximação costeira- 1)

A média, erro padrão e o fator "pior" ou "maior" (média dos três maiores valores) desses recursos foram calculados para cada imagem, resultando em 30 recursos. Por exemplo, o campo 3 é o raio médio, o campo 13 é o raio SE, o campo 23 é o pior raio.

São 569 instâncias rotuladas nesta base, divididas entre 357 benignas e 212 malignas.

V. RESULTADOS OBTIDOS E ANÁLISE

Os resultados apresentados ao longo desta seção foram obtidos através da implementação em Python do perceptron multicamadas. Utilizou-se a biblioteca Keras, que permite a criação de protótipos de redes neurais a partir de comandos de alto nível.

A. Teste inicial

Foi utilizada uma camada de entrada com 8 neurônios (função de ativação *relu*). Duas camadas escondidas com 16 neurônios cada (função de ativação *relu*). Camada de saída com função de ativação *tanh* e dois neurônios (um para cada classe a ser classificada: maligno ou benigno). Foi utilizado batch de tamanho 30 e otimizador *RMSprop*.

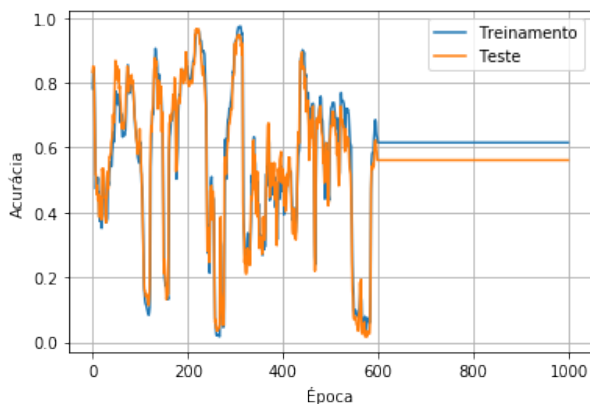


Figura 6. Acurácia para o teste inicial.

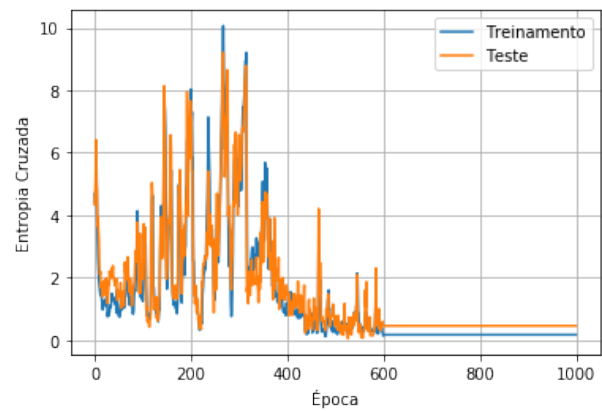


Figura 7. Função custo para o teste inicial.

Os resultados finais foram acurácia de 0,615 e função custo 0,177 no treinamento e acurácia de 0,561 e função custo de 0,462 no teste. Não foram resultados muito empolgantes.

B. Aumentando o número de neurônios

A tentativa agora será de aumentar o número de neurônios para 64 em cada camada escondida e 32 na camada de entrada, mantendo os demais parâmetros iguais.

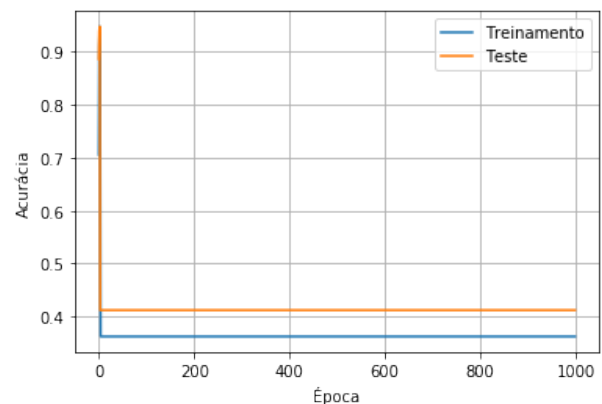


Figura 8. Acurácia para o treinamento 2, de aumento de número de neurônios.

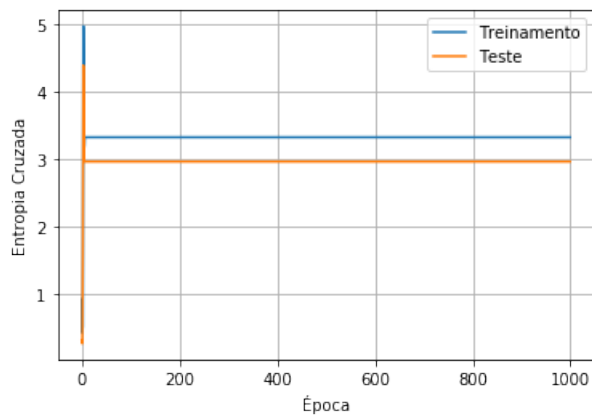


Figura 9. Função custo para o treinamento 2, de aumento de número de neurônios.

Como pode ser visto pelas Figuras 8 e 9, os resultados continuaram muito ruins e a rede nem está aprendendo. O problema não parece ser com o número de neurônios, mas talvez com as funções de ativação.

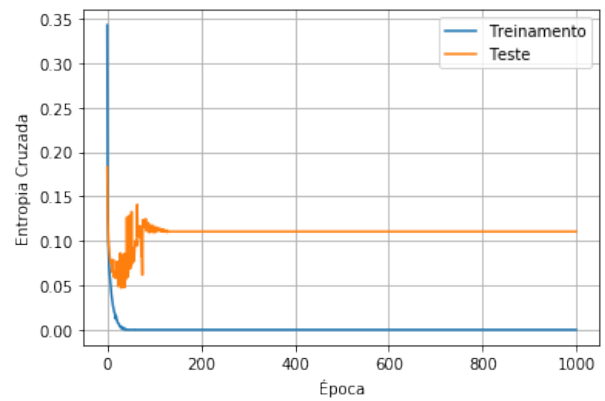


Figura 11. Função custo entropia cruzada com a função de ativação *softmax*.

Como pode ser visto pelas Figuras 10 e 11, agora sim os resultados foram muito satisfatórios: acurácia 1 e entropia cruzada de $1,19 \times 10^{-7}$ no treinamento e acurácia 0,965 e entropia cruzada de 0,11 no teste.

D. Reduzindo o número de neurônios

C. A função de ativação *softmax*

Mudando a função de ativação de saída para *softmax* e mantendo os demais parâmetros do teste anterior (32 neurônios na entrada e 64 em cada uma das duas camadas escondidas).

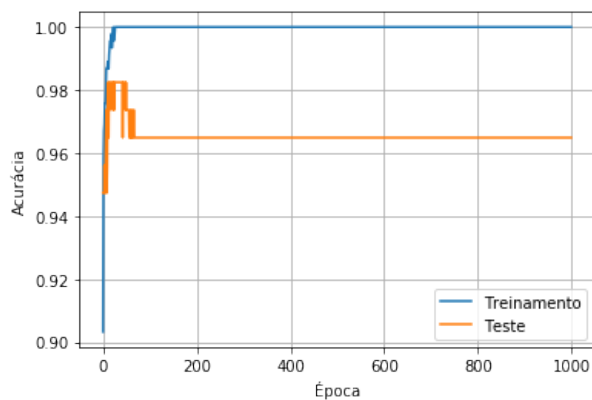


Figura 10. Acurácia com a função de ativação *softmax*.

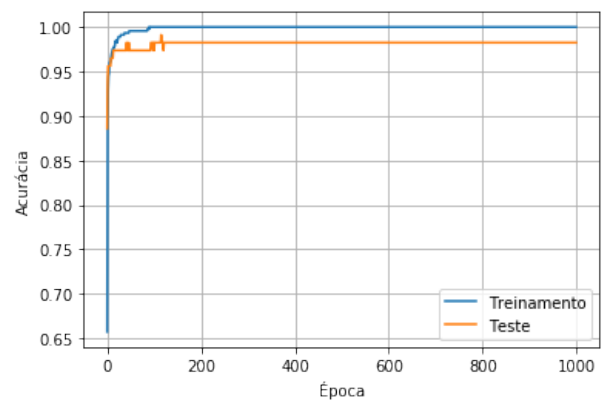


Figura 12. Acurácia para softmax na saída e menos neurônios.

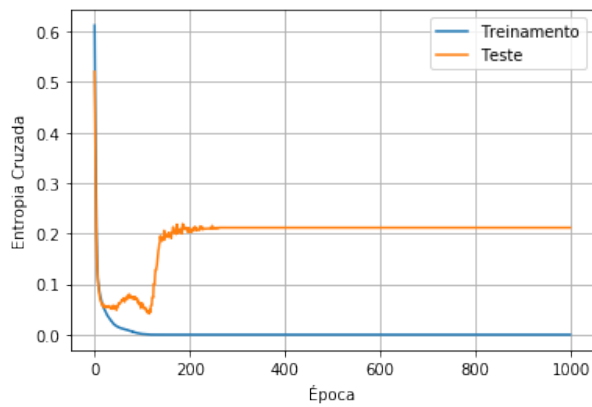


Figura 13. Função custo entropia cruzada para softmax na saída e menos neurônios.

Como pode ser visto pelas Figuras 12 e 13, o resultado foi muito bom. Ao final se obteve acurácia 1 e entropia cruzada de $1,19 \times 10^{-7}$ no treinamento e acurácia 0,982 e entropia cruzada de 0,212 no teste. A acurácia melhorou com menos neurônios.

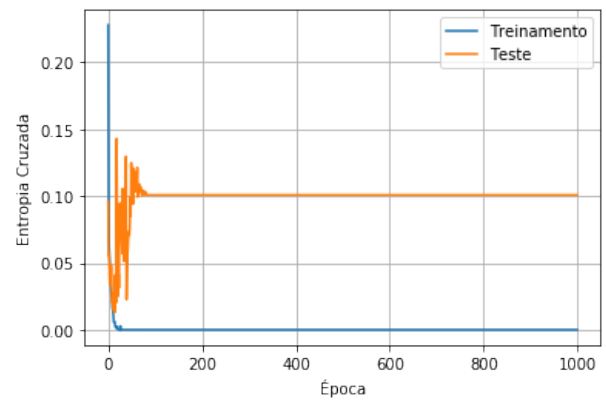


Figura 15. Função custo entropia cruzada para softmax na saída e menos neurônios.

Os resultados finais foram acurácia 1 e entropia cruzada de $1,19 \times 10^{-7}$ no treinamento e acurácia 0,974 e entropia cruzada de 0,101 no teste. De fato, o aumento do número de neurônios não parece melhorar o aprendizado da rede. Como não houve melhora, volta-se a utilizar 8 neurônios na camada de entrada e 16 neurônios em cada uma das duas camadas escondidas.

E. Aumentando o número de neurônios.

Para confirmar a tese de que menos neurônios melhorou a rede, o próximo teste será com 128 neurônios em cada uma das camadas escondidas e 64 neurônios na camada de entrada, mantendo os demais parâmetros.

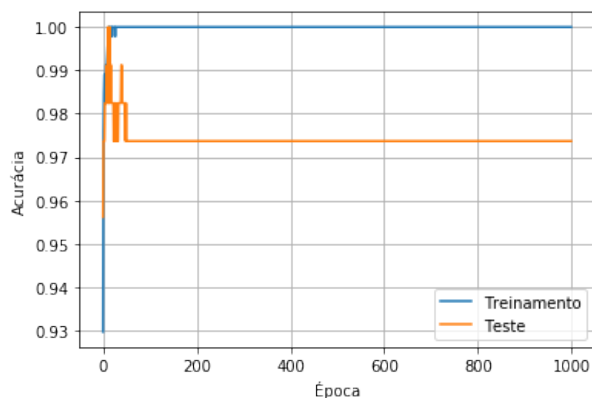


Figura 14. Acurácia para softmax na saída e mais neurônios.

F. Aumentando o número de camadas escondidas

A próxima tentativa será de colocar mais uma camada escondida, totalizando assim uma camada de entrada com 8 neurônios e 3 camadas escondidas com 16 neurônios cada, mantendo os demais parâmetros.

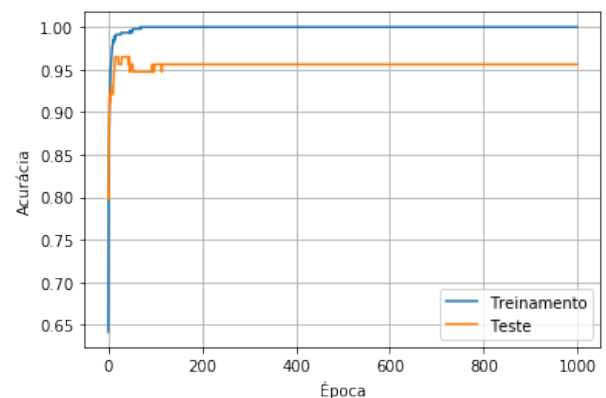


Figura 16. Acurácia para rede com mais uma camada escondida.

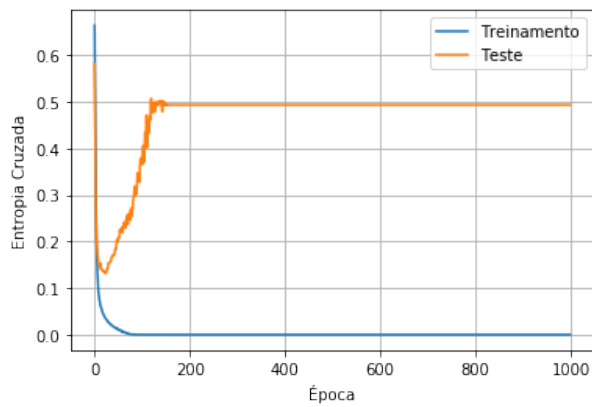


Figura 17. Função custo entropia cruzada para rede com mais uma camada escondida.

Os resultados finais foram acurácia 1 e entropia cruzada de $1,19 \times 10^{-7}$ no treinamento e acurácia 0,956 e entropia cruzada de 0,493 no teste. Essa tentativa piorou o aprendizado da rede tanto em acurácia quanto na função de custo final.

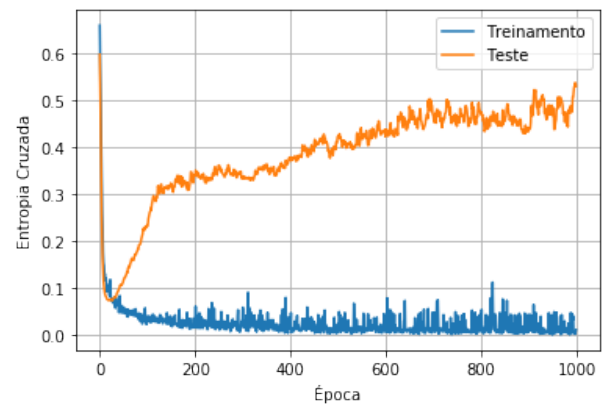


Figura 19. Função custo entropia cruzada para rede com *dropout*.

Os resultados finais foram acurácia 0,99 e entropia cruzada de 0,01 no treinamento e acurácia 0,956 e entropia cruzada de 0,529 no teste. Essa tentativa piorou o aprendizado da rede tanto em acurácia quanto na função de custo final, além disso os resultados foram muito instáveis. Desta forma, *dropout* não é bom para este pequeno conjunto de dados.

G. Adicionando Dropout

Serão mantidos agora a camada de entrada com 8 neurônios e duas camadas escondidas com 16 neurônios cada. Será adicionado *dropout* a uma taxa de 20% entre a camada de entrada e a primeira camada intermediária.

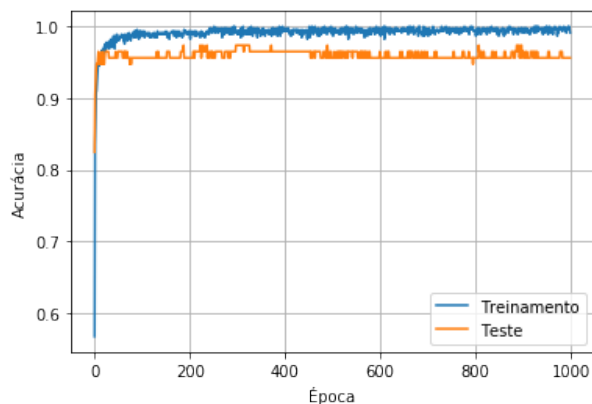


Figura 18. Acurácia para rede com *dropout*.

H. Utilizando regularização l1 e l2

A próxima tentativa será de utilizar regularização l1 e l2 de 0,01 em cada camada escondida, retirando o *dropout* que havia sido colocado e mantendo os demais parâmetros.

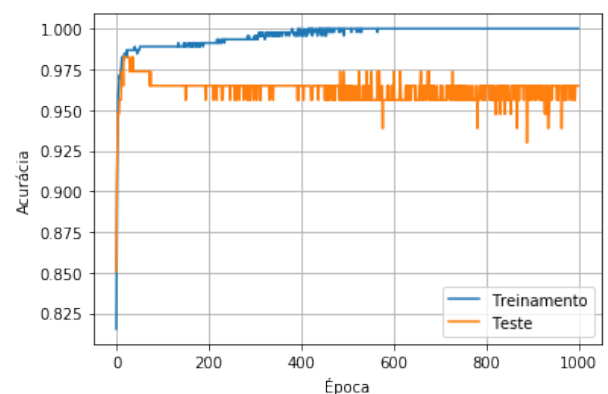


Figura 20. Acurácia para rede com regularização l1 e l2.

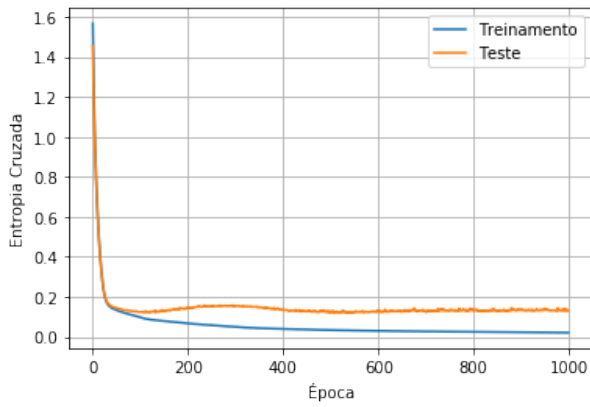


Figura 21. Função custo entropia cruzada para rede com regularização $l1$ e $l2$.

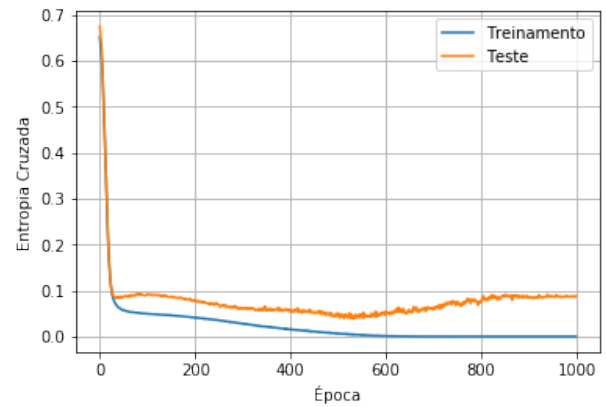


Figura 23. Função custo entropia cruzada com camadas de entrada e escondidas utilizando função de ativação *sigmoid*.

Os resultados finais foram acurácia 1 e entropia cruzada de 0,021 no treinamento e acurácia 0,965 e entropia cruzada de 0,13 no teste. Essa tentativa piorou o aprendizado da rede tanto em acurácia quanto na função de custo final em relação ao treinamento sem aplicação de regularização $l1$ e $l2$, além disso os resultados foram muito instáveis. Desta forma, esta técnica também não será utilizada.

Os resultados finais foram acurácia 1 e entropia cruzada de $1,2 \times 10^{-7}$ no treinamento e acurácia 0,991 e entropia cruzada de 0,0879 no teste. Essa tentativa melhorou o aprendizado da rede tanto em acurácia quanto na função de custo final em relação ao treinamento utilizando função de ativação *relu* nas mesmas camadas.

I. Alteração da função de ativação das demais camadas

A próxima tentativa será alterar a função de ativação nas camadas de entrada e escondidas para *sigmoid*, mantendo os demais parâmetros.

J. Alteração do tamanho de batch

Para o próximo teste, será alterado o tamanho de batch de 30 para 20 e o número de épocas será aumentado para 3000, mantendo os demais parâmetros.

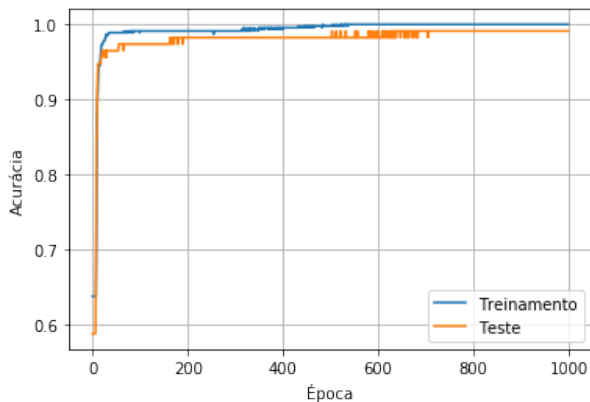


Figura 22. Acurácia com camadas de entrada e escondidas utilizando função de ativação *sigmoid*.

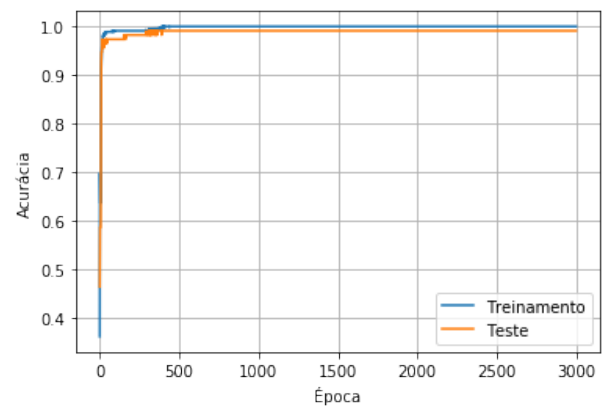


Figura 24. Acurácia para batch de tamanho 20.

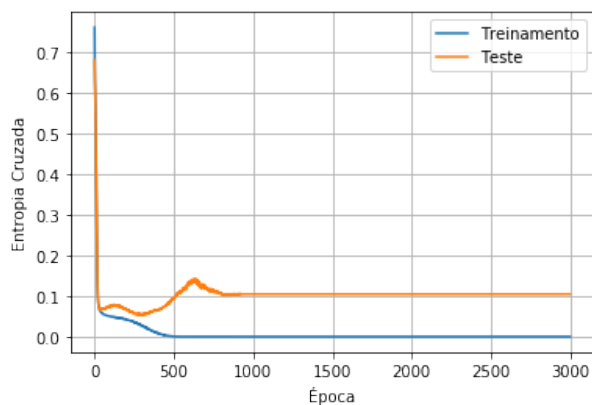


Figura 25. Função custo entropia cruzada para batch de tamanho 20.

Os resultados foram iguais ao treinamento anterior, à exceção da entropia cruzada no treinamento, que aumentou para 0,105. O número de épocas não precisa ser maior do que 1000, já que a rede se estabilizou bem antes disso. Como os resultados foram muito satisfatórios, com a rede praticamente não errando mais, estes parâmetros obtidos foram tratados como suficientes para o propósito deste trabalho. Desta forma, foi gerada a matriz de confusão para a rede com os melhores parâmetros.

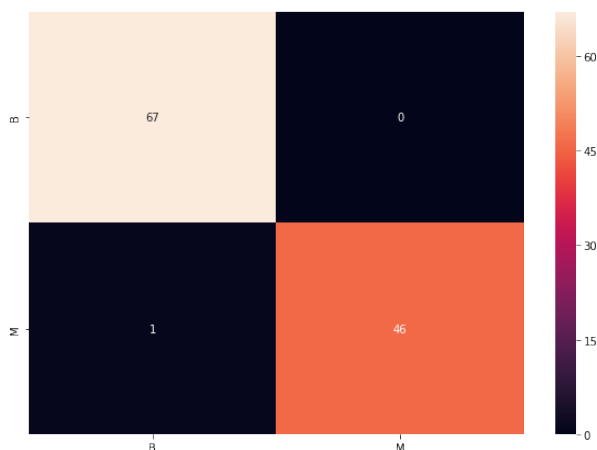


Figura 26. Matriz de confusão da última rede treinada.

Pode ser visto na matriz de confusão (Figura 26) que a rede errou apenas uma análise, classificando um tumor que era maligno como benigno.

VI. CONCLUSÃO

O relatório teve como proposta a realização de algoritmos para classificação de imagens de massas tumorais. Os resultados obtidos se mostraram absolutamente viáveis como meio para obtenção de melhoria no processo de diagnóstico de câncer de mama. A discriminação dos padrões de malignidade e benignidade das massas é um processo que necessita alta precisão.

Nesta avaliação de dados, chegamos ao resultado de apenas um falso negativo. Apesar de ser um erro estatisticamente

baixíssimo, um diagnóstico como esse pode ocasionar em falta de tratamento ou atraso no início do mesmo, aumentando exponencialmente a chance de complicações à saúde do paciente. Esta é uma área que minimizar os erros nem sempre é suficiente, deve-se buscar formas de se anular as taxas de erros.

O relatório cumpre com o que foi pedido em avaliação final na disciplina Inteligência Artificial. Entretanto, a ressalva fica por conta de que para se obter um material de conclusão mais preciso seria adequada a realização de experimentos com imagens de outras bases, e se fosse possível, com um número maior de imagens, além da realização de testes com outras técnicas de aprendizado de máquina, como abordagens de aprendizagem profunda, ou outros classificadores como Random Forests e AdaBoost.

REFERÊNCIAS

- [1] Ivan N , Spatti, Danilo H. Flauzino, Rogério A, Redes Neurais Artificiais para engenharia e ciências aplicadas, Artliber, 2010.
- [2] S. Haykin, Neural Networks and Learning Machines, Pearsey, New York, Third Edition, 2009.
- [3] M. Nielsen, Neural Network and Deep Learning; e-book <http://neuralnetworkanddeeplearning.com>.
- [4] Data Science Academy. Deep Learning Book, 2019. Disponível em: <http://www.deeplearningbook.com.br/>. Acesso em: 1 Julho, 2019.
- [5] Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [[https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+\(Diagnostic\)](https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic))]. Irvine, CA: University of California, School of Information and Computer Science.