Universidade do Minho

Mestrado Integrado em Engenharia Informática

Computação Natural

Redes Neuronais Artificiais em Exames Clínicos

Luís Gomes (A78701)

Joel Rodrigues (A79068)

Elisa Valente (A79093)

13 de Abril de 2019

Conteúdo

Figuras

1 Introdução

O presente documento, elaborado no âmbito da unidade curricular de Computação Natural, pretende apresentar as decisões tomadas na realização deste trabalho prático.

A utilização de modelos de *Machine Learning* para a resolução de problemas do quotidiano tem vindo a ganhar uma maior importância e visibilidade, devido aos resultados positivos que têm apresentado nos últimos anos. Modelos como as Redes Neuronais Artificiais (RNA's) são muito utilizados em casos que envolvam a classificação baseada num conjunto de atributos e onde existam dados suficientes para o treino e avaliação do modelo criado. A precisão destes modelos está muito dependente da arquitetura da rede pelo que os parâmetros que a definem devem ser otimizados de modo a conseguir obter os melhores resultados possíveis.

Existem várias técnicas que efetuam a otimização de parâmetros de modelos de classificação, sendo que o seu principal objetivo passa por treinar o modelo várias vezes com diferentes configurações e escolher o conjunto de hiper-parâmetros que obtém uma melhor classificação segundo uma métrica predefinida. Os Algoritmos Genéticos (AG) enquadram-se neste grupo de técnicas de otimização e utilizam conceitos da biologia relacionados com o cruzamento e mutações genéticas, bem como a evolução das espécies e seleção natural introduzida por Charles Darwin.

Neste trabalho é proposto um problema onde é necessária a classificação de massas detetadas em mamografia como `benignas` ou `malignas`. Para isto é utilizado um modelo *deep learning*, mais concretamente uma rede neuronal artificial que é otimizada com recurso a algoritmos genéticos.

2 Descrição do problema

A mamografia ou mastografia é um exame médico utilizado para a deteção de anomalias no tecido mamário através da utilização de raio-x para a obtenção de imagens radiográficas. Com o objetivo de prever se uma determinada massa é ou não maligna propõe-se a criação de um modelo que, utilizando os dados fornecidos por este tipo de exame, obtenha uma percentagem aceitável de acerto, quer na classificação de massas benignas, mas especialmente no que às malignas diz respeito.

Os dados disponíveis e a respetiva representação são os seguintes:

- **Age:** patient's age in years (integer)
- **Shape:** mass shape: round=1 oval=2 lobular=3 irregular=4 (nominal)
- **Margin:** mass margin: circumscribed=1 microlobulated=2 obscured=3 ill-defined=4 spiculated=5 (nominal)
- **Density:** mass density high=1 iso=2 low=3 fat-containing=4 (ordinal)
- **Severity:** benign=0 or malignant=1 (binominal)

3 Análise e preparação do *dataset*

De forma a obter um conhecimento mais detalhado sobre o domínio dos dados disponíveis foi efetuada uma análise, visualizando as gamas de valores de cada atributo, assim como a contagem de campos em falta. Como se pode visualizar na tabela 1, o número de dados em falta não é muito significativo dada a diferença das ordens de grandeza para o número total de ocorrências no *dataset* (961 linhas). Para além disto foi efetuada uma análise da distribuição destes valores ao longo do conjunto de dados, e visto que estes se encontravam aleatoriamente repartidos, a melhor opção foi descartar as linhas com valores nulos.

Atributo	**Número de NaN**
Age	5
Shape	31
Margin	48
Density	76
Severity	0

Tabela 1: Tabela representativa do número de valores nulos de cada atributo.

De acordo com a figura 1, podemos também verificar que a gama de valores dos atributos difere em larga escala de coluna para coluna, podendo estes valores afetar o desempenho do modelo criado. Assim, estes dados foram padronizados e a figura 2 é a representação dos dados que serão utilizados.

	age	shape	margin	density
0	67.0	3.0	5.0	3.0
2	58.0	4.0	5.0	3.0
3	28.0	1.0	1.0	3.0
8	57.0	1.0	5.0	3.0
10	76.0	1.0	4.0	3.0

Figura 1: Dados não padronizados.

	age	shape	margin	density
0	0.765804	0.174460	1.395631	0.240313
2	0.151666	0.979883	1.395631	0.240313
3	-1.895458	-1.436386	-1.158927	0.240313
8	0.083429	-1.436386	1.395631	0.240313
10	1.379941	-1.436386	0.756992	0.240313

Figura 2: Dados padronizados.

4 Treino e validação de modelos de aprendizagem

Para o treino e validação dos modelos de aprendizagem é efetuada uma separação dos dados como representado na figura 3. Como o nome indica, o conjunto *training data* é utilizado para o treino do modelo e o conjunto *validation data* será utilizado para obter o *score* final da rede neuronal após a otimização de hiper-parâmetros, sendo estes dados totalmente novos para a rede. Estes dados são separados aleatoriamente, com uma distribuição de percentagens `validação-treino` de `5-95`.

Figura 3: Divisão do *dataset* em dados de treino e validação.

Para melhorar a confiança na avaliação da rede é importante a utilização de técnicas como o *Kfold cross-validation* que nos permitam obter pontuações mais precisas com diferentes subconjuntos de treino e teste para uma mesma configuração de rede. Após a obtenção de todos os *scores* é efetuada a sua média aritmética representativa da avaliação do modelo. A figura 4 ilustra o treino e avaliação de uma rede neuronal artificial.

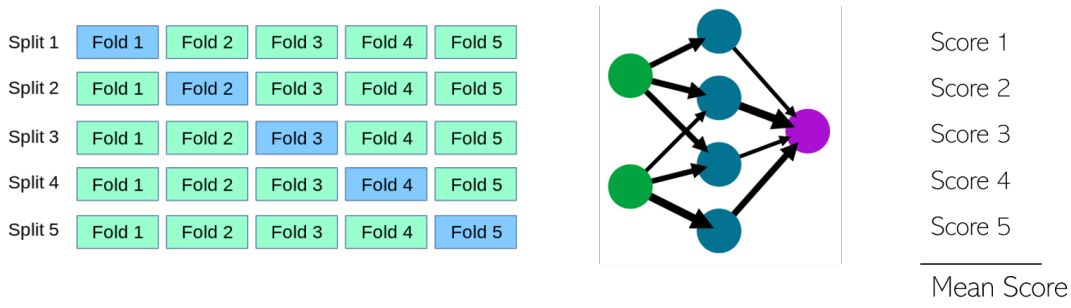

Figura 4: Modelo de aprendizagem.

A função utilizada como *score* tem em conta o contexto em que o problema se enqua-

dra, dando especial atenção ao acerto de casos em que o paciente apresenta uma massa maligna. Assim, é adicionado um novo critério à função de avaliação, o *recall*, que indica a percentagem de casos malignos corretamente previstos. A equação 1 apresenta essa função.

$$Score = (Accuracy * 0.8) + (Recall * 0.2) \tag{1}$$

5 Otimização de parâmetros do modelo - Algoritmos Genéticos

Para a otimização de parâmetros do modelo optou-se por algoritmos genéticos em lugar de qualquer outro algoritmo de otimização tradicional. Após a comparação de diferentes métodos, verifica-se que para o problema em questão, os algoritmos genéticos apresentam algumas vantagens[1]:

- Baseiam-se na codificação do conjunto de soluções possíveis, e não nos próprios parâmetros de otimização;
- Os resultados são apresentados como uma população de soluções e não como uma solução única;
- Não necessitam de nenhum conhecimento derivado do problema, apenas de uma forma de avaliação do resultado;
- Usam transições probabilísticas e não regras determinísticas.

5.1 Representação do problema

Nesta técnica de otimização é necessário representar o problema específico num cromossoma composto por genes. Cada gene representa uma característica do problema, contendo um valor na gama dos valores possíveis para essa característica. Neste caso o problema pode ser representado da seguinte forma:

[1]Citando `Goldberg, David E. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning`: *1. GAs work with a coding of the parameter set, not the parameters themselves. 2. GAs search from a population of points, not a single point. 3. GAs use payoff (objective function) information, not derivatives or other auxiliary knowledge. 4. GAs use probabilistic transition rules, not deterministic rules.*

- **Gene 1** - Número de camadas intermédias de uma rede. Este valor varia entre 1 e 10.
  Valores possíveis: [1,2,...,10]
- **Gene 2** - Número de nodos em cada camada. No caso em estudo todas as camadas apresentam o mesmo número de nodos. Estes são codificados em potências de 2 com valores entre 2 e 128.
  Valores possíveis: [1,2,...,7]
- **Gene 3** - Taxa de aprendizagem. Este parâmetro denota a agressividade na atualização dos pesos da rede neuronal. Varia entre 10^{-8} a 10^{-2}.
  Valores possíveis: [2,3,...,8]
- **Gene 4** - Função de ativação. As funções de ativação utilizadas são a *Relu* e a *Sigmoid*, codificadas em 0 e 1 respetivamente.
  Valores possíveis: [0,1]

5.2 Iterações do algoritmo genético

Em cada iteração do algoritmo genético existe uma população que representa um conjunto de redes neuronais, através dos cromossomas definidos na secção 5.1. Para a execução das várias iterações do algoritmo genético existem 4 etapas fundamentais, treino, seleção, cruzamento e mutação, que se encontram de seguida explicadas.

5.2.1 Obtenção de *Fitness*

Para cada uma das redes é efetuado o processo de *kfold cross-validation*, como visto anteriormente na secção 4. Os dados são escolhidos sequencialmente sem fator aleatório, para que os *scores* obtidos dependam unicamente da arquitetura da rede e não da variabilidade dos dados envolvidos. De notar que a aleatoriedade necessária é garantida pela divisão inicial.

Como se pode ver na figura 5, cada um dos indivíduos da população é classificado de acordo com uma função de *fitness* que corresponde à média dos *scores*, cujo valor é apresentado na secção 4 (equação 1).

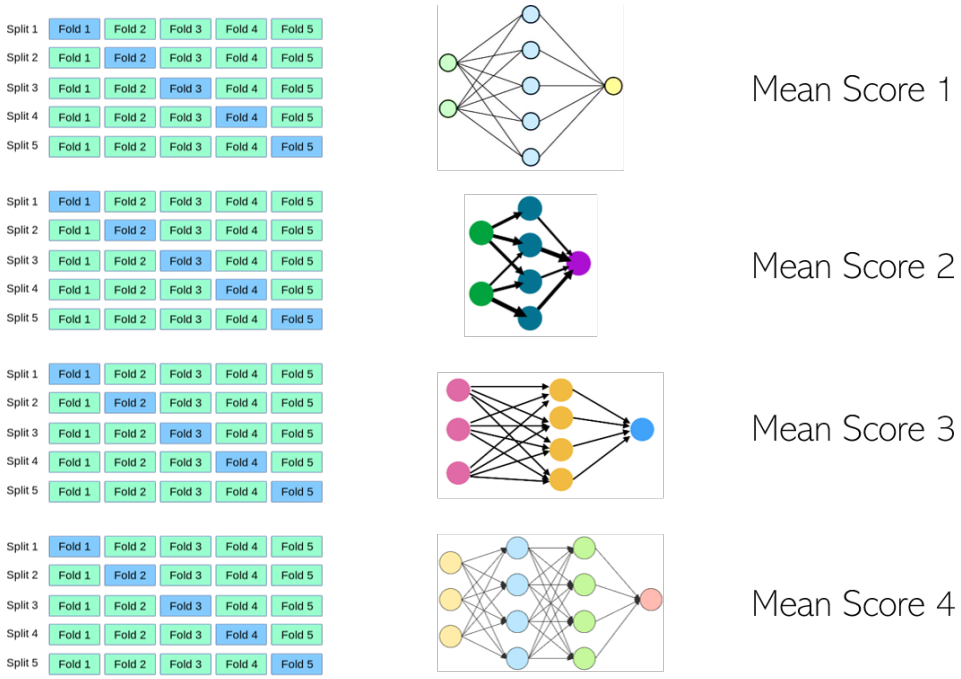

Figura 5: Representação de uma iteração do algoritmo genético.

5.2.2 Seleção

De forma a apurar os indivíduos com maior valor de aptidão, e posteriormente passar para a fase de reprodução dos mesmos, recorre-se à seleção elitista. Nesta são selecionados os N melhores indivíduos da população intermediária, ou seja, aqueles que apresentem maior *score*. Neste caso, o N considerado foi `Pop Size` / 2. Na figura 6 encontram-se representados a laranja os cromossomas que passam para a geração seguinte, sendo o *Pop Size* = *4* e *N* = *2*.

Figura 6: Seleção dos melhores cromossomas para a geração seguinte.

5.2.3 Cruzamento

O operador seguinte à seleção é o cruzamento e, tal como nos métodos reprodução sexuada dos seres vivos, este processo tem como objetivo a transferência do material genético dos pais para os descendentes. A principal motivação para realizar este operador em cada iteração do algoritmo genético é gerar novos indivíduos na população que sejam ainda mais aptos, neste caso, produzam melhores *scores*. No problema em questão, decidiu-se usar um ponto de corte único para efetuar o cruzamento, sendo assim, para cada pares de progenitores, são gerados 2 descendentes, herdando metade das características do pai e metade da mãe. A figura 7 é representativa do processo de cruzamento para um determinado par de indivíduos.

[[2, 4, 0.0000001, Relu], [4, 16, 0.001, Relu], [2, 4, 0.001, Relu], [4, 16, 0.0000001, Relu]]

Figura 7: Cruzamento entre os melhores cromossomas.

5.2.4 Mutação

Após várias iterações do algoritmo genético é comum que a população resultante apresente características muito semelhantes e, como tal, deixe de evoluir. Deste modo, é realizada a mutação de um gene arbitrário num dos cromossomas descendentes selecionado aleatoriamente. Esta mutação tem como objetivo introduzir uma variabilidade genética na população, impedindo que a evolução populacional fique estagnada. A figura 8 apresenta uma mutação ocorrida num cromossoma filho. Neste caso, o gene afetado foi o da função de ativação, tendo esta passado de `Relu` para `Sigmoid`.

[[2, 4, 0.0000001, Relu], [4, 16, 0.001, Relu], [2, 4, 0.001, Relu], [4, 16, 0.0000001, Relu]]

[[2, 4, 0.0000001, Relu], [4, 16, 0.001, Relu], [2, 4, 0.001, Sigmoid], [4, 16, 0.0000001, Relu]]

Figura 8: Mutação num cromossoma filho.

6 Resultados Obtidos

Após o término do processo de otimização com 6 gerações e uma população de 12 cromossomas, a arquitetura que apresenta melhor valor de *fitness* é a seguinte:

- **Numero de camadas intermédias:** 1
- **Número de nodos em cada camada:** 32
- **Taxa de aprendizagem:** 10^{-2}
- **Função de ativação:** *Relu6*

Esta arquitetura apresenta uma classificação de 0.817.

Após a escolha da melhor rede neuronal, procede-se ao seu treino com a totalidade dos dados de treino divididos no ínicio do processo de *Machine Learning* (95% de todo o *dataset*). Com a rede treinada, é possível obter uma avaliação geral, submetendo os dados de validação e comprovando os resultados. A figura 9 representa este processo, onde é visível a arquitetura final e o valor da *accuracy* (80.95%).

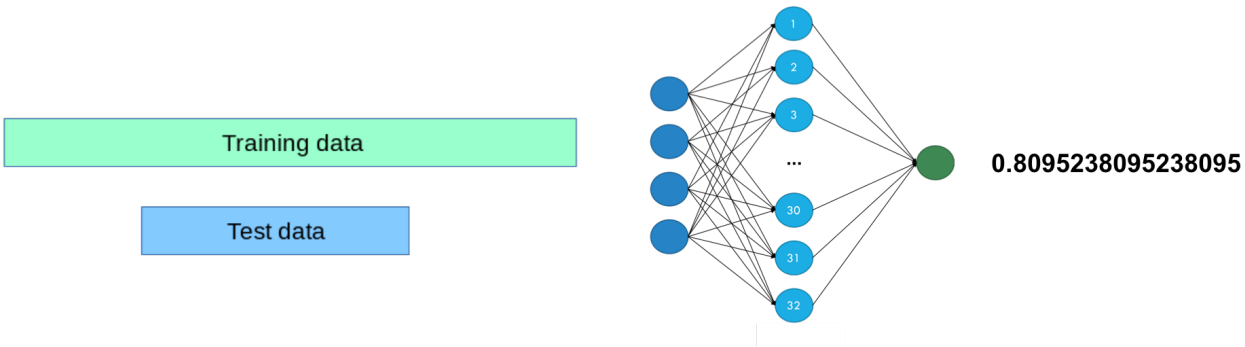

Figura 9: Validação final da melhor arquitetura obtida.

A matriz de confusão associada a esta arquitetura é a mostrada na tabela 2.

	Benigno Previsto	Maligno Previsto
Benigno Real	18	6
Maligno Real	2	16

Tabela 2: Matriz de confusão do modelo final.

Sendo o *recall* uma medida que é importante para o caso em estudo, vemos que este se aproxima de 89% (16/(16+2)).

7 Conclusão

A aplicação de redes neuronais artificiais combinada com a otimização dos seus parâmetros com algoritmos genéticos revela-se como uma forma eficaz de resolver problemas. Estes algoritmos propõem-se a encontrar, após algumas iterações, a arquitetura de rede que mais se adequa ao problema.

No caso estudado, os resultados obtidos são bastante satisfatórios, uma vez que a rede obtida consegue indicar corretamente se uma massa em tecido mamário é maligno ou benigno em cerca de 80% dos casos (*accuracy*). Para além disto, espera-se que o modelo preveja corretamente cerca de 90% dos casos malignos (*recall rate*).

Apesar do sucesso deste método, é de notar o pesado poder computacional necessário para a execução dos algoritmos. De facto, a partir da 3ª geração do algoritmo genético verifica-se que não existe uma melhoria significativa nos melhores indivíduos da população, estagnando perto dos 0.8. A partir desta geração apenas ocorrem melhorias nos restantes indivíduos, existindo uma convergência em todos eles para arquiteturas semelhantes e consequentemente também dos *scores*. Assim, na última geração vê-se que nenhum indivíduo tem *score* abaixo de 0.8 e as arquiteturas apresentam uma das possíveis combinações de parâmetros:

- **Número de camadas:** 1 ou 5
- **Número de nodos:** 32 ou 64
- **Taxa de aprendizagem:** 10^{-2}
- **Função de ativação:** ReLu

Como forma de melhorar o trabalho realizado no futuro, foi pensada a inclusão dos *steps* e *batch size* no processo de otimização, acrescentando dois genes a cada cromossoma. Para além disto, caso a rede fosse implementada num cenário real, seria interessante realizar um treino com a totalidade dos dados, na tentativa de aumentar a qualidade das previsões.