Deep Learning Interpretável

Gonçalo Almeida[pg47212], Leonardo Marreiros[pg47398], Maria Sofia Marques[pg47489], and Pedro Fernandes[pg47559]

Universidade do Minho, Braga, Portugal

Resumo As redes neuronais profundas são bem conhecidas pelo seu excelente desempenho em lidar com várias tarefas de *machine learning* e inteligência artificial. Contudo, devido à sua natureza de caixa negra sobreparametrizada, é muitas vezes difícil entender os resultados de previsão de modelos profundos. Nos últimos anos, muitas ferramentas de interpretação foram propostas para explicar ou revelar as formas pelas quais os modelos profundos tomam decisões.
Neste artigo seguimos essa linha de pesquisa começando por abordar preocupações na definição de interpretabilidade no contexto de *machine learning* e a sua importância. Técnicas que se comprometem a ajudar na interpretabilidade destes modelos como *feature importance*, *SHapley Additive xPlanations* (SHAP), *occlusion* e GRAD-CAM e as suas aplicações na deteção de cancro da pele, deteção de COVID-19 e na deteção de *Malware* são exploradas. Terminamos este artigo com uma breve análise dos desafios que este assunto engloba.

Palavras-Chave: *Deep Learning* interpretável · *Machine learning* · cancro da pele · covid-19 · *malware* · interpretação imagem

1 Introdução

Modelos *deep learning* tem alcançado um desempenho notável numa variedade de campos, desde reconhecimento visual, processamento de linguagem natural, *reinforcement learning*, a sistemas de recomendação, onde estes modelos produziram resultados comparáveis e, em vários casos, superiores a especialistas humanos.

Apesar do reconhecimento da importância de *machine learning* (ML), a importância da interpretação torna-se clara para evitar consequências catastróficas. Devido à sua natureza de sobreparametrização (envolvendo mais de milhões de parâmetros além de centenas de camadas), muitas vezes é difícil entender os resultados de previsões de modelos profundos. Explicar os seus comportamentos continua um desafio devido à sua natureza opaca ou de caixa negra (*black box)*, isto é, sem qualquer conhecimento do seu funcionamento interno. A falta de interpretabilidade levanta sérias questões sobre a confiança de modelos *deep* em aplicações de alto risco onde erros seriam catastróficos tais como condução autónoma [1], medicina [2], operações militares [3] e justiça criminal [4].

Apesar de muitas ferramentas de interpretação (visualizações, linguagem natural, equações matemáticas) já terem sido propostas para explicar a forma

como modelos ML tomam decisões, de um ponto de vista científico ou social, a explicação dos comportamentos destes modelos ainda está em progresso. Isso tem levado a considerável confusão sobre a noção de interpretabilidade. É incerto o que significa interpretar algo, e como selecionar um método de interpretação para um determinado problema/público.

Além de abordar este problema, iremos explorar também em concreto algumas técnicas que tem sido propostas e desenvolvidas para tornar modelos ML mais interpretáveis como *feature importance*, *SHapley Additive xPlanations* (SHAP), *occlusion* e GRAD-CAM e as suas aplicações.

Finalmente, desafios técnicos como a fidelidade ambígua e o controlo da taxa de erro são explorados sucintamente.

2 *Deep Learning* Interpretável

2.1 O que é Interpretabilidade?

Segundo o dicionário Priberam, *interpretação* é o "sentido em que se toma o que se ouve ou o que se lê, e que se julga ser o verdadeiro". [5] No contexto de sistemas *machine learning*, é adicionado um ênfase em fornecer explicações a seres humanos, isto é, explicar ou apresentar termos compreensíveis para um ser humano.

Embora explicação possa ser um termo mais intuitivo que interpretabilidade, resta ainda clarificar então, o que é uma explicação? Tal como muitos outros termos idênticos, uma definição formal de explicação permanece indescritível. Na área da psicologia, Lombrozo [6] argumenta que "explicações são mais do que uma preocupação humana - são centrais aos nossos sentidos de compreensão, e a moeda através da qual trocamos crenças"e menciona que questões como o que constitui uma explicação, o que torna uma explicação melhor que outra, como surge uma explicação e quando são procuradas explicações, estão apenas a começar a ser abordadas.

De facto, quando Lee Sedol, ex-jogador profissional do jogo de tabuleiro Go, enfrentou AlphaGo, um programa que utiliza inteligência artifial para jogar este jogo, descreveu que havia jogadas sobre-humanas e afirmou que apesar da Humanidade ter passado já milhares de horas a jogar Go, era claro que ainda não o compreendiam este jogo de todo. No entanto, apesar de saber que há algo de mais profundo no jogo, o programa não diria o que era, apenas derrotava todos os jogadores humanos. Havia algo que o AlphaGo estava a aprender mas não era possível extrair este conhecimento. A possibilidade de interpretar métodos de *deep learning* como este, deveria dar-nos uma ideia do porquê das máquinas terem uma performance tão superior.

Da mesma forma, com carros autónomos, gostaríamos de saber, quando ocorre um acidente, o porquê do carro não ter virado ou o porquê de não ter travado, qual foi a interpretação. Para depois ser possível atribuir culpas ao *machine learning*, ou às condições, ou à companhia de seguros, ou a outra entidade.

No diagnóstico de doenças, gostaríamos de saber o porquê de um modelo ML ter feito uma certa classificação de um tumor como uma leucemia, por exemplo.

Ou o porquê de certos medicamentos serem prescritos, o porquê do diagnóstico, o porquê de um paciente apenas ter uma determinada probabilidade de sobrevivência. Essa explicação pode ajudar em como fazer melhores diagnósticos no futuro, como questionar um diagnóstico de um paciente e como melhor compreender as características do modelo para depois treinar uma melhor seleção de características, um melhor classificador ou até para ser capaz de ensinar aos médicos como fazer melhores previsões.

Não há duvida que o impacto social de ML está a começar a ser considerado seriamente. Em 2018 a União Europeia lançou uma regulação, GDPR (*General Data Protection Regulation*) que inclui um "direito a explicação"aos cidadãos sujeitos a decisões automatizadas. [7]

De forma geral, apesar do reconhecimento do valor de ML, modelos *deep learning* (e não só) são essencialmente caixas negras em termos de interpretabilidade. São demasiado difíceis para um humano entender ou proprietários, para que não se possa entender o seu funcionamento interno. Os modelos caixa negra geralmente prevêem a resposta certa pelo motivo errado, levando a um excelente desempenho em treino mas um baixo desempenho na prática. [13]

O objetivo geral é perceber como e por que estes modelos se comportam da forma como se comportam extraindo informação interpretável das caixas negras.

2.2 Porquê Interpretabilidade?

Verificar que o modelo funciona como esperado: Decisões erradas podem ser extremamente dispendiosas. Novamente no contexto de carros autónomos, seria preferível que o modelo que os guia explicitasse as condições específicas sob as quais não foi capaz de tomar uma decisão do que suspender a atividade de todos estes veículos até ser encontrada a razão para o sucedido.

Do mesmo modo, a falta de interpretabilidade nos modelos de ML pode potencialmente ter consequências adversas ou mesmo fatais. Por exemplo, considerando o trabalho de Caruana *et al.* [14] na construção de classificadores para identificar pacientes com pneumonia com baixo ou alto risco de morte no hospital. Uma rede neuronal, que pode ser vista como uma caixa negra em termos de interpretabilidade, foi determinada como o melhor classificador para o problema. Investigação sobre este problema revelou que um dos principais preditores era o histórico de asma do paciente, uma doença pulmonar crónica. O modelo previa que dada asma, o paciente teria um menor risco de morte no hospital quando admitido por pneumonia. Na verdade, o oposto é verdadeiro - pacientes com asma estão em maior risco de complicações graves e sequelas, incluindo morte, de uma doença pulmonar infeciosa como pneumonia. Isto acontecia porque os pacientes com asma receberam cuidados mais oportunos e de maior acuidade do que os pacientes sem asma, induzindo numa maior vantagem de sobrevivência. Assim sendo, no caso de classificação errada, idealmente, deveria ser possível atribuir isso a características específicas dos dados ou do modelo.

Melhorar/Depurar o classificador: Normalmente, em ML comum existem os dados, o modelo, as previsões e algum tipo de erro de generalização ou risco. Com

ML interpretável, existe algum tipo de interpretabilidade extraído da caixa negra (modelo) que é inspecionado via humano e com isto é capaz de melhorar os dados ou o modelo (Figura 1).

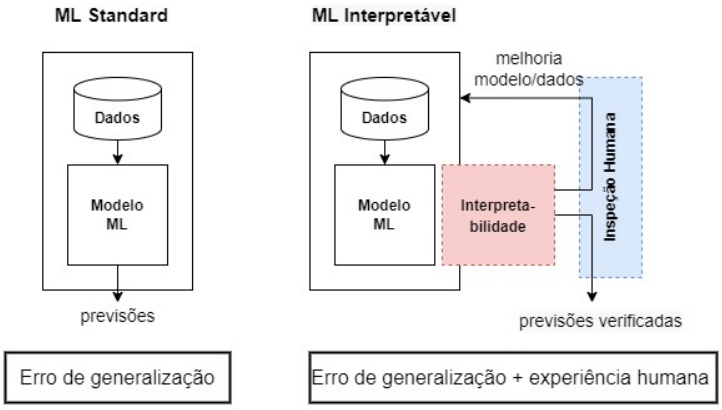

Figura 1. ML Standard vs ML Interpretável

Fazer novas descobertas: Ao saber aquilo que o modelo está a "ver"numa imagem de um tumor por exemplo, é possível depois treinar os médicos para conseguirem reconhecer esses padrões.

Direito à explicação: Para algoritmos que são treinados de forma discriminativa, isto é, se por exemplo a raça de uma pessoa tem peso na sua contratação para um emprego, com ML interpretável seria possível saber o porquê do modelo estar a fazer essas decisões e depois eliminar esses preconceitos originando um algoritmo limpo.

ML interpretável permite, portanto, interrogar, perceber, depurar e até melhorar o sistema de apredizagem. Permite aos utilizadores finais avaliarem o modelo antes de qualquer decisão ser tomada. Ao explicar o raciocínio por trás das previsões, sistemas de ML interpretável dão aos utilizadores razões para aceitar ou rejeitar previsões e recomendações.

2.3 Técnicas de *deep learning* interpretável

Feature importance Hoje em dia é possível obter-se previsões com um elevado grau de precisão através de algoritmos de aprendizagem. Infelizmente, as regras de decisão são dificilmente acessíveis aos humanos e não podem ser facilmente usadas para obter *insights* sobre o domínio do problema. *Feature importance* foi uma técnicas concebidas com o objetivo de selecionar variáveis, sacrificando algum poder preditivo para interpretabilidade presuntiva.

Selecionar os atributos apropriados para os modelos tem uma importância muitas vezes ignorada, uma vez que, dados inúteis resultam em *bias* que atrapalha os resultados finais da aprendizagem da máquina.

No fundo, esta técnica calcula uma pontuação para todos os *inputs* de um determinado modelo, sendo que essas pontuações representam a "importância" de cada *input* . Uma pontuação mais alta significa que o recurso específico terá um efeito maior no modelo que está a ser usado para prever uma determinada variável.

Esta técnica é extremamente útil e interpretável no sentido em que a importância do *feature* permite entender a relação entre os *inputs* e a variável final, como também ajuda a entender quais os *features* que são irrelevantes para o modelo. Isso não torna apenas o modelo mais simples, mas também acelera o funcionamento do modelo, melhorando o seu desempenho .

Occlusion A oclusão é uma técnica simples que procura entender quais as partes de uma imagem que são mais importantes para a classificação de uma rede profunda. Podemos medir a sensibilidade de uma rede à oclusão em diferentes regiões dos dados usando pequenas perturbações nos mesmos. A sensibilidade de oclusão permite compreender de uma forma interpretável quais os recursos de uma imagem a rede usa para fazer uma classificação específica e fornecer informações sobre os motivos pelos quais uma rede pode classificar incorretamente uma imagem.

Esta técnica é conseguida então através da perturbação de pequenas áreas do *input* substituindo-as por uma máscara de oclusão, normalmente um quadrado cinzento. A máscara move-se pela imagem e a mudança na pontuação de probabilidade para uma determinada classe é medida em função da posição dessa mesma máscara. Este método permite assim destacar quais as partes da imagem que são mais importantes para a classificação, ou seja quando essa parte da imagem é ocluída, a pontuação de probabilidade para a classe prevista cairá drasticamente.

Um mapa de oclusão pode ainda mostrar se o objeto a ser classificado na imagem é o correto. Se a rede não está a produzir os resultados esperados, um mapa de oclusão pode ajudar a entender qual o motivo. Por exemplo, se a rede estiver fortemente focada em outras partes da imagem que não o objeto que se pretende, isso sugere que a rede aprendeu os *features* errados.

A oclusão pode ainda servir para comparar quais partes da imagem a rede identifica como evidência para diferentes classes. Isso pode ser útil nos casos em que a rede não confia na classificação e dá pontuações semelhantes para várias classes. Nesses caos um mapa de oclusão para cada uma das principais classes permite examinar os resultados da oclusão com maior resolução, reduzindo o tamanho da máscara (*MaskSize*) e o passo (*Stride*). Um *Stride* menor leva a um mapa de maior resolução, mas com maior tempo computacional e maior uso da memória. Um *MaskSize* menor ilustra detalhes menores, mas pode levar a resultados mais ruidosos.

Podemos concluir então que esta técnica é bastante útil e altamente interpretável no sentido em que permite compreender quais os principais fatores de decisão, como também ajuda a entender onde o modelo pode estar a falhar.

Shap values *(SHapley Additive exPlanations)*, proposto por Lloyd Shapley em 1953, é uma abordagem teórica do campo da teoria dos jogos cooperativos, cujo objetivo é explicar o output de qualquer modelo de *machine learning*, medir o contributo de cada "jogador"para o respetivo "jogo".

Supondo um contexto em que um grupo "n"entidades ao trabalhar juntos obtiveram um lucro "L"e desejam separar os lucros de forma justa e não igual para todos. Para distribuir o lucro justamente, é necessário medir a contribuição de cada membro, ou seja, o valor Shapley de cada uma das entidades. Para tal, dado uma entidade "A", a diferença entre o lucro gerado quando o "A"está presente é calculada em relação ao lucro gerado quando esta entidade está ausente, essa diferença é a contribuição marginal do membro dado para a coalizão atual. Este cálculo é feito para todos os subgrupos (ou coalizões) que podem ser gerados onde a entidade "A"se encontra presente. A média das diferenças obtidas (a contribuição marginal média) é o valor de Shapley. Em palavras simples, o valor de Shapley de cada jogador é a contribuição marginal média de uma instância de um recurso entre todas as coalizões possíveis.

Esta técnica é bastante utilizada pois permite fornecer explicabilidade global (consiste em entender a estrutura no geral de como um modelo faz um decisão) e local (corresponde em entender como um modelo faz decisões para uma determinada situação). Por fim, ao contrário de outros métodos, SHAP permite uma maior flexibilidade uma vez que pode ser usado para qualquer modelo baseado em árvore e não apenas para modelos lineares e/ou de regressão logística.

Grad-CAM *Gradient-weighted Class Activation Mapping*, abreviado por Grad-CAM é uma técnica popular para visualizar o que a rede neuronal está à "procura", podendo assim provar se a uma rede neuronal está a olhar para os padrões corretos presentes no modelo. Grad-CAM tem como principal característica visualizar separadamente cada classe presente numa imagem.

A ferramenta Grad-CAM torna possível partir de uma imagem inteira (e não apenas uma imagem com anotações de local explicitas) e determinar a localização de um respetivo objeto. Por exemplo, numa imagem que esteja presente vários animais é possível identificar um animal em especifico. Esta ferramenta também pode ser combinada para visualizações de espaço e pixel existentes num determinado objeto, sem necessitar rótulos de nível de pixel para treino.

Em linguagem simplificada, Grad-CAM produz uma mapa de calor (*heatmap*) que destaca as áreas relevantes de uma relativa imagem com base nos gradientes do alvo em questão (um objeto em especifico escolhido pelo agente) da camada convolucional final. Posteriormente são utilizados os "mapas de *features*"da camada final e compara-se todos os canais nesse *feature* com o gradiente da classe em relação ao canal. Isto permite saber com que intensidade a imagem *input* ativa diferentes canais pela importância de cada canal em relação à sua

classe. Esta ferramenta não requer nenhum pré-treino ou mesmo alteração na arquitetura existente.

De uma forma geral, Grad-CAM explora a informação espacial que está preservada entre as camadas convolucionais com o intuito de perceber quais partes de uma dada imagem são necessárias para uma decisão correcta de classificação.

3 Aplicações

3.1 Deteção de cancro da pele

Face ao *Deep Learning* Interpretável, uma possível utilização do mesmo encontra-se na área da saúde, mais precisamente no campo da patologia digital. Foi colocado em teste no artigo [19] a segmentação e classificação multiclasse de cancro de pele não melanoma. As formas mais comuns deste tipo de cancro incluem carcinoma basocelular (BCC) que compreende aproximadamente 60% de todos os diagnósticos e carcinoma espinocelular (SCC) compreendendo mais 30%. Um outro tipo de cancro de pele não melanoma designa-se carcinoma intraepidérmico (IEC).

Os dados usados para treino e teste provieram de uma coleção pré-existente de amostras de cancro da pele, fornecidas pelo *MyLab Pathology*, onde posteriormente, cada lâmina (amostra) foi anotada à mão por um patologista para indicar qual secção de tecido era mais representativa da classe de cancro. O conjunto de dados reunidos a partir das amostras representam ainda os fatores indicadores da existência/prevalência do cancro de pele não melanoma na população.

Um dos principais objetivos deste projeto consistiu então em caracterizar o tecido das amostras classificando-o em 12 classes dermatológicas significativas. No fundo, procura-se encontrar um método interpretável para realizar classificação de imagens de modo a ser possível obter um diagnóstico correto de cancro de pele não melanoma. Este projeto visa abranger os 3 cancros da pele mais comuns, que correspondem a 90% dos diagnósticos.

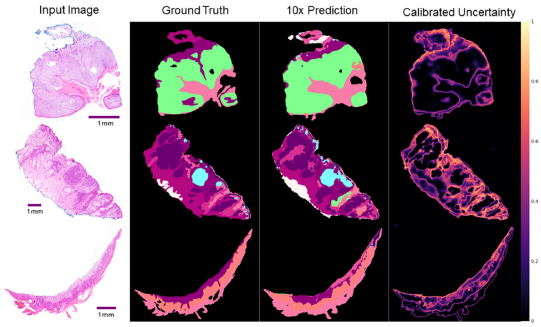

Figura 2. Segmentação semântica de seções de tecido inteiro. [19]

Na figura 2 encontra-se um exemplo de explicação como a rede classifica um segmento numa das 12 possíveis classes de cancro juntamente com o modo que calcula o grau de incerteza calibrado. É relevante acrescentar que as diferentes cores observadas nas amostras identificam as diferentes classes de cancro examinadas. Na primeira coluna encontramos imagens reais de segmentos de tecido. Na segunda coluna podemos observar aquilo que seria esperado observar após diagnóstico. Ou seja, o local no tecido que se encontra contaminado. A terceira coluna apresenta a previsão das segmentações de imagens inteiras da rede 10x para casos representativos de SCC (*Punch, Shave, Excision*), BCC (*Excision, Shave, Shave*) e IEC (*Shave, Excision, Shave*) respetivamente. Por fim a última coluna representa o mapa de incerteza calibrado dos segmentos analisados.

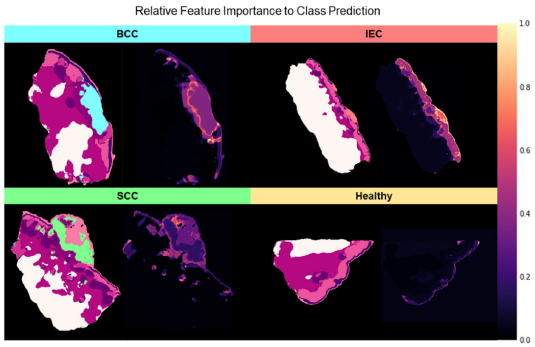

Figura 3. Medição da importância relativa de cada classe no classificador CNN. [19]

Na figura 3 é apresentada uma outra forma de visualizar os resultados. Os gradientes da imagem do lado direito do painel revelam quais as regiões diagnosticamente importantes e, portanto, quais classes de tecido são importantes para uma determinada previsão de classe.

Após a visualização de ambas as figuras 2 e 3 torna-se evidente que este método funciona satisfatoriamente em vários tipos de cancro.

Os mapas de incerteza são ainda apresentados na figura 2, onde mostram quais as partes da imagem que a rede considerou mais complexo. Estes mapas de incerteza permitem de uma forma extremamente interpretável avaliar o desempenho da rede. Para além disso, a apresentação destes mapas fornecem um sistema de apoio à decisão permitindo ao patologista um meio de localizar áreas que requerem mais atenção.

A qualidade das segmentações proporcionou ainda a oportunidade de realizar cálculos automáticos de depuração da margem cirúrgica. A figura 4 mostra um exemplo de uso de técnicas clássicas de processamento de imagem para detectar e medir distâncias até as margens cirúrgicas, onde 90% das previsões estavam dentro de 0,27mm da margem verdadeira.

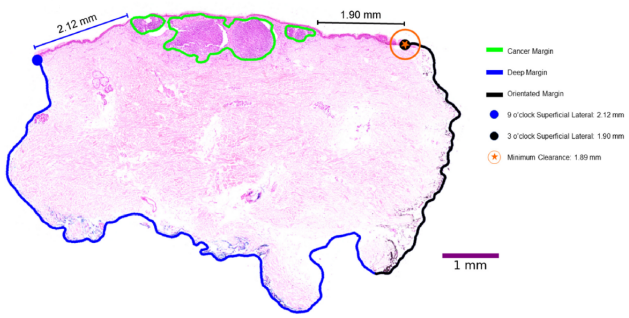

Figura 4. Exemplo possível apresentação da avaliação automática da desobstrução da margem cirúrgica [19]

De forma encorajadora, estes resultados demonstram que este método não só apresenta resultados satisfatórios para diagnóstico, como também pode fornecer uma desobstrução de margem cirúrgica robusta que, em teoria, se poderia estender para realizar outras tarefas de rotina.

É dito ainda neste artigo que, se o número de classes fosse estendido para incluir tecido vascular e nervoso, as distâncias do cancro a essas regiões poderiam ser calculadas facilmente. Isso forneceria suporte à decisão/avaliação automática da invasão linfovascular e perineural.

3.2 Deteção de COVID-19

Uma outra aplicação para o *Deep Learning* Interpretável, também na área da saúde, foi a deteção de casos de covid-19 colocado em teste no artigo [15]. Muitos modelos antes deste foram propostos para detetar este vírus, no entanto nenhum deles era interpretável, algo que nesta área é muito importante, uma vez que no que toca à saúde humana, é necessário explicar o porquê dos resultados obtidos. Neste artigo foi introduzido um modelo de *Deep Learning* interpretável constituído por dois modelos deste tipo.

Os dados usados para treino e teste foram obtidos pela junção de dois *datasets*: um disponível no *Kaggle* e outro no *Github*. Em ambos apenas foram aproveitadas imagens raio-X frontais de pessoas normais, pacientes com covid-19 e pacientes com pneumonia. Como o número de imagens relativamente a pacientes covid-19 era muito inferior às outras duas categorias, foi necessário balancear os dados para que o modelo funcionasse da melhor forma, para tal foram incluídas copias de imagens de pacientes de covid-19.

O objetivo era encontrar um método interpretável para realizar classificação de imagens de modo a ser possível afirmar o porquê de uma imagem ter sido classificada de uma determinada forma. Para prever a classe de uma imagem teste, o modelo do artigo [15] calcula os *scores* de similaridade entre partes prototipadas de imagens de cada classe com partes de imagens de teste através de uma função de distância. Estes *scores* em seguida serão multiplicados por

uma matriz de *weights* para estabelecer conexões positivas entre protótipos e *logits* de classes corretas e usar conexões nulas ou negativas em caso de classes incorretas.

O modelo resultante foi construído sobre os modelos de redes neuronais convolucionais amplamente utilizados: VGG-16, VGG-19, ResNet-34, ResNet-152, DenseNet-121, ou DenseNet-161. Estes modelos são considerados os modelos base para o modelo final. Na figura 5 é possível observar que o modelo apresenta as camadas de convolução de qualquer um dos modelos acima referidos, com a adição de uma camada de dimensão 1x1. Estas camadas de convolução c são seguidas por uma camada de protótipo pp(camada de convolução generalizada) e uma camada totalmente conectada m com matriz de peso wtm e sem *bias*. A ReLU é a função de ativação usada para todas as camadas de convolução.

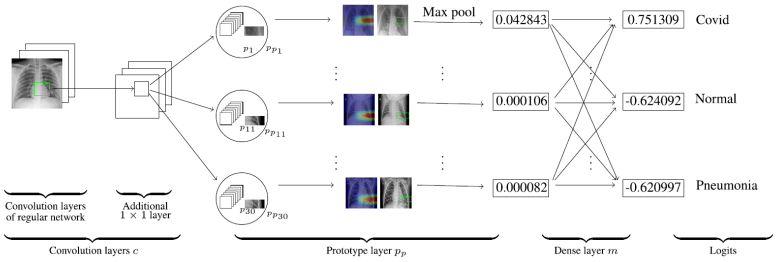

Figura 5. Arquitetura do modelo *Gen-ProtoPNet* [15]

Na figura 6 encontra-se um exemplo de explicação de como a rede avalia um paciente na classe covid-19. Na primeira coluna está uma imagem de teste de um paciente na classe de covid-19. A coluna seguinte contém a mesma imagem só que com um retângulo verde a delimitar uma determinada parte dos pulmões. A quarta coluna contém o retângulo da imagem original à qual corresponde o retângulo da segunda coluna na mesma linha. A terceira coluna representa simplesmente a projeção dessa seleção da quarta coluna. As colunas seguintes correspondem respetivamente aos *scores* de similariedade e com os *weights* que pertencem cada um a uma matriz. Fazendo multiplicações de linhas da matriz de *weights* correspondentes à classe covid-19 com o respetivo *score* de similaridade obtém-se o *logit* para a classe de covid-19.

3.3 Deteção de *Malware*

Uma terceira aplicação desta técnica aqui presente pode ser encontrada no artigo [16]. Desta vez deixamos aplicações relacionadas com saúde e passamos para o ramo da tecnologia. *Software* malicioso, também conhecido como *Malware* afeta qualquer dispositivo que possua sistema operativo, desde os tradicionais computadores, até aos mais recentes *smartphones* e *wearables*. Neste caso, o alvo do projeto foi relativamente à deteção de *Malware* em dispositivos *android*.

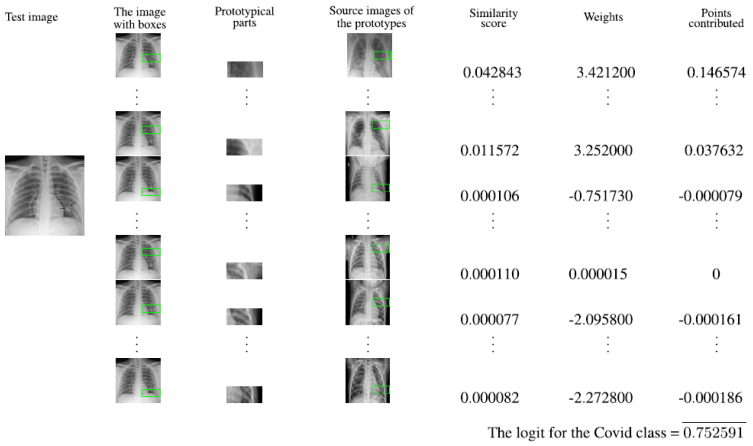

Figura 6. Explicação da rede com uma imagem de um paciente com Covid-19 [15]

O modelo da análise experimental contou com mais de oito mil amostras divididas em sete famílias de *software* diferentes, seis das quais maliciosas e uma última de aplicações confiáveis. Estes agrupamentos por família foram depois pré-processados e transformados em imagens. Da aplicação *android* é extraído o ficheiro de executável *dex* e em seguida a sequência de *bytes* é agrupada em *unsigned integers* de 8 *bits* que pode posteriormente ser visto em pixeis a preto e branco.

A metodologia deste projecto passou essencialmente por seis passos diferentes. Estes passos podem ser divididos em duas secções distintas, os passos habituais para treinar um modelo e os passos explicativos. Na primeira parte estão incluídas partes como o *dataset* explicado em detalhe anteriormente, o modelo *Deep Learning* usado, seguido da sua divisão em treino e teste. A segunda secção é aquela que permite o modelo ser interpretável sendo constituído por três passos, a geração do *heatmap* usando *Grad-CAM*, os *heatmaps* acumulativos e a avaliação da aprendizagem.

1. **Geração do *heatmap*:** um subconjunto de amostras é selecionado a partir do conjunto de dados, sendo este classificado pelo modelo que identifica as partes da imagem que mais servem de classificação para uma determinada classe. O resultado desta classificação é um *heatmap* para a própria classe.
2. ***Heatmaps* acumulativos:** todos os *heatmaps* de uma determinada família de classificação são agrupados calculando um único *heatmap* que possui a média para cada pixel dos *heatmaps* que lhe deram origem.
3. **Avaliação de aprendizagem:** são analisadas as semelhanças de todos os *heatmaps* de modo a melhorar a informação presente no *heatmap* acumulativo.

Na Figura 7, podemos observar esquematicamente a metodologia referida.

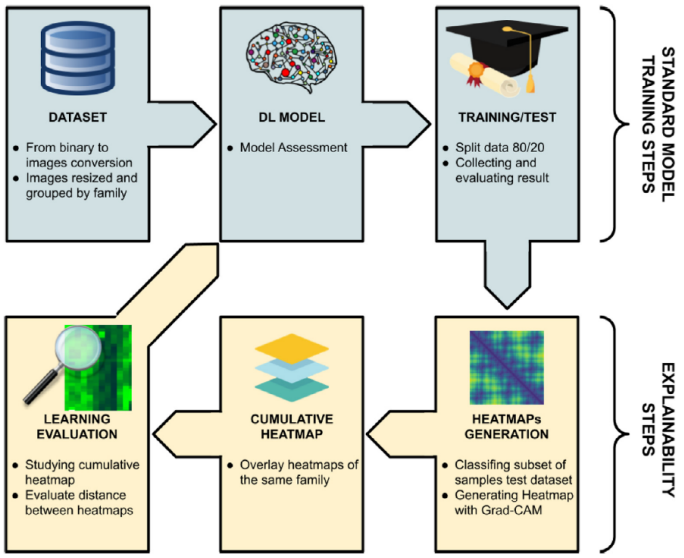

Figura 7. Metodologia aplicada no artigo [16]

Já relativamente à figura 8 o *heatmap* consegue captar os padrões mais importantes da imagem de um *software* malicioso. O *heatmap* acumulativo permite a um especialista de segurança analisar os resultados obtidos do modelo. Este é formado pela junção de 50 amostras e compõe resultados relativos a uma determinada família de *malware*. De modo a evitar uma classificação errada deve-se evitar a existência de dois *heatmaps* semelhantes para classes diferentes. Apenas quando *heatmaps* acumulativos apresentem padrões minimamente afincados, podem ser considerados úteis para identificar um determinado *malware*. Estes *heatmaps* são bastante úteis uma vez que com eles, um analista pode fazer *reverse engineering* à imagem resultada, chegando assim às secções de código potencialmente malicioso podendo assim analisá-lo com maior cuidado.

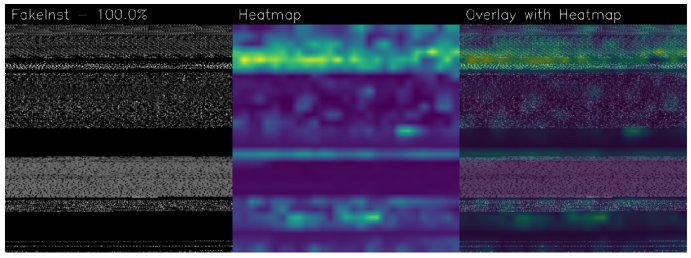

Figura 8. Comparação entre a imagem de *malware* e o *heatmap* [16]

4 Desafios

Nos últimos anos testemunhamos avanços consideráveis em deep learning interpretável. Apesar desse progresso, ainda existem desafios a superar ou mesmo reconhecer adequadamente certos obstáculos conceituais fundamentais. De acordo com o artigo [20] são três desafios conceituais que são amplamente ignorados pelos autores nesta área. Sendo estes os seguintes:

1. Ambiguidade em relação ao seu verdadeiro alvo;
2. Desrespeito pelas taxas de erro e teste;
3. Ênfase no produto sobre o processo.

O primeiro desafio corresponde a fidelidade ambígua. Este recai na questão: a quem devem as explicações algorítmicas ser fiéis, ao modelo de destino ou ao processo de geração de dados?

É de conhecimento geral que o mais importante para qualquer ferramenta IML (*interpretable machine learning*) é a precisão e as explicações que "*são verdadeiras, ou pelo menos aproximadamente corretas*". No entanto, estes objetivos são bastante sub especificados. Se tivermos em consideração dois tipos de métodos de atribuição de recursos diferentes é reconhecido que cada um é leal a um alvo diferente. Logo, temos que colocar a questão - qual dos dois tipos é o mais adequado - e para ter uma resposta tem-se que ter em consideração informações pragmáticas relativas ao contexto, o nível de abstração e o propósito da investigação subjacente.

Em linguagem mais corrente e dado um exemplo da vida real, que conjunto de circunstâncias físicas explicam o facto de alguém ter um doença rara y apesar de não apresentar sintomas aparentes?. Há uma ambiguidade inerente no objetivo mais óbvio e incontroverso do IML. Este alguém quer uma explicação algorítmica que seja verdadeira, precisa e fiéil, mas fiel a quê? Ao modelo ou o sistema? Devemo-nos preocupar mais com a função de diagnóstico que prevê que esse alguém tem a doença rara y, ou aos factos biológicos que constituem condições de verdade para a previsão?

Com o fim de compreender melhor o desafio, é feita uma breve revisão sobre dois desafios bem conhecidos inter-relacionados que complicam os esforços para inferir e quantificar os efeitos causais:

– **O problema da indução:** A ideia básica é que a inferência de observações particulares para generalizações universais depende de alguma suposição de uniformidade natural. Por exemplo, para existir o salto entre "Todos os gatos observados são pretos"para "todos os gatos são pretos"é pressuposto que, lá por que a primeira afirmação é corroborada num determinado tempo e espaço, vale para qualquer espaço e/ou tempo. Porém, os céticos argumentam que tal premissa não pode ser justificada pela razão e/ou experiência. Tal como é vidente, isto acarreta obstáculos para qualquer causa que deseje passar de apenas correlação, estruturas mais profundas são inobserváveis em princípio.

– **Possíveis fatores de confusão:** Supondo que existe dependência estatística persistente entre duas variáveis, X e Y, esta só pode ser explicada por uma de três circunstâncias: (a) X causa Y; (b) Y causa X; ou (c) uma terceira variável Z causa X e Y, o que a torna um confundidor. Esta pode induzir numa correlação entre X e Y e classificar erroneamente a dependência entre os dois como uma instância (a) ou (b). Por exemplo, a eficácia de um estudo clínico está diretamente relacionada com fatores demográficos. Caso os grupos de tratamento e controle difiram substancialmente ao longo de variáveis relevantes, como idade ou sexo isso pode significar que o estudo em questão não é fiável porque conta com fatores que podem alterar o resultado final, uma variável Z. No entanto, não é possível controlar todos os fatores de confusão devido a restrições inevitáveis, tais como orçamento, instrumentos e/ou imaginação.

A seguinte questão que se coloca em causa incide no controlo da taxa de erro, uma vez que a grande maioria dos métodos IML nem se preocupam em quantificar taxas de erro esperadas. Isso torna impossível submeter explicações algorítmicas a testes severos, como é exigido para qualquer hipótese científica.

Por exemplo, os algoritmos como LIME ou SHAP, têm *outputs* facilmente inteligíveis mas isso significa necessariamente que as suas explicações devem ter o mesmo peso, ou algumas são mais confiáveis do que outras? Como podemos ter certeza de que estes algoritmos não produziram estimativas instáveis ou selecionaram os recursos errados?

Argumenta-se assim que testes severos são a chave para garantir a confiabilidade das explicações algorítmicas. É assim, por exemplo, que se passa a confiar em teorias científicas - submetendo-as impiedosamente a inúmeros testes com taxas de erro quantificáveis.

Infelizmente, a maioria dos autores de IML ainda não tomaram conhecimento desta questão. Até que testes severos sejam incorporados ao IML, o campo deixará de cumprir os padrões de rigor científico exigidos para ampla adoção e confiança do utilizador.

Por último, existe a questão sobre processo versus produto. As abordagens atuais tratam predominantemente as explicações como entregas estáticas e únicas. De facto, explicações bem-sucedidas são mais um processo do que um produto, estas exigem refinamentos dinâmicos e iterativos entre vários agentes. Explicações contra factuais e com base num respetivo caso geram exemplos destinados a esclarecer as previsões de um modelo. Em cada caso, o *output* será um produto, ou seja, uma entrega estática que é computada de forma definitiva.

Neste tópico, defende-se que uma forma eficaz de pensar em possíveis explicações é como um processo, isto é, uma troca iterativa entre (pelo menos) dois agentes envolvidos num certo tipo de investigação causal. Para além desta explicação se assemelhar mais com a vida real, também são mais simples de compreender por parte de um elemento de fora.

Explicações iterativas permitem que o agente não experiente obtenha uma visão mais completa do modelo em questão e que adquira um conhecimento mais amplo sobre a matéria. Assim que reconhecemos que diferentes agentes

solicitarão explicações algorítmicas com diferentes motivações, expectativas, e crenças, em vez de criar uma solução de tamanho único, a abordagem dialógica permite que o agente inquiridor guie a discussão para melhor satisfazer as suas necessidades e curiosidades.

5 Conclusão

Modelos *black box* estão cá para ficar. Instituições públicas e privadas já contam com ML para executar funções básicas e complexas com maior precisão e eficiência do que humanos. *Datasets* crescentes e *hardware* cada vez melhor, em combinação com avanços contínuos na ciência da computação e estatística, garantem que estes métodos só se tornarão cada vez mais omnipresentes nos próximos anos.

Não há razões para acreditar que estes algoritmos se tornarão mais transparentes ou inteligíveis, pelo menos não sem o esforço contínuo e explicito de investigadores no campo promissor de *machine learning* interpretável.

É fácil substimar o quão difícil é convencer alguém a usar um modelo ML na prática, e a interpretabilidade é um fator chave. Apesar de técnicas para ML interpretável estarem a avançar rapidamente, alguns dos principais desafios permanecem sem solução, e as soluções futuras são necessárias para promover ainda mais o progresso deste campo.

A literatura que tem surgido neste campo pode ser algo confusa com terminologias como "explicabilidade"e "interpretabilidade"usadas de forma aparentemente igual, apesar de não ser este o caso. No mínimo esperamos ter introduzido alguns princípios fundamentais, e ter coberto várias áreas importantes do campo e a forma como se relacionam entre si e com problemas reais.

Referências

1. Jinkyu Kim, John Canny *Interpretable Learning for Self-Driving Cars by Visualizing Causal Attention* (2017)
2. Watson, D., Krutzinna, J., Bruce, I. N., Griffiths, C. E. M., McInnes, I. B., Barnes, M. R., et al. *Clinical applications of machine learning algorithms: Beyond the black box* BMJ, 364, 446–448. (2019)
3. Gunning, D. *Explainable artificial intelligence (XAI)* (2019)
4. Cynthia Rudin, Berk Ustun *Optimized Scoring Systems: Towards Trust in Machine Learning for Healthcare and Criminal Justice*
5. "interpretação", in Dicionário Priberam da Língua Portuguesa [em linha], 2008-2021, https://dicionario.priberam.org/interpreta%C3%A7%C3%A3o [consultado em 16-03-2022].
6. Lombrozo T *The structure and function of explanations. Trends in cognitive sciences* 10(10):464–470 (2006)
7. Art. 21 GDPR Right to object, https://gdpr-info.eu/art-21-gdpr/ [consultado em 16-03-2022].
8. Grad-CAM: Visual Explanations from Deep Networks, https://glassboxmedicine.com/2020/05/29/grad-cam-visual-explanations-from-deep-networks/ [consultado em 16-03-2022].

9. Grad-CAM Reveals the Why Behind Deep Learning Decisions, https://www.mathworks.com/help/deeplearning/ug/gradcam-explains-why.html [consultado em 16-03-2022].
10. Understand Network Predictions Using Occlusion, https://www.mathworks.com/help/deeplearning/ug/understand-network-predictions-using-occlusion.html [consultado em 16-03-2022].
11. Understanding Feature Importance and How to Implement it in Python, https://towardsdatascience.com/understanding-feature-importance-and-how-to-implement-it-in-python-ff0287b20285 [consultado em 16-03-2022].
12. Intro to SHAP values in Python, https://deepnote.com/@joshzwiebel/Intro-to-SHAP-values-in-Python-fetVcsOYSA-aIRG-bmWF7g [consultado em 16-03-2022].
13. Ashoori, M. and Weisz, J. D. *In AI We Trust? Factors That Influence Trustworthiness of AI-infused Decision-Making Processes*
14. Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. *Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 1721-1730 (2015)
15. G. Singh, K.-C. Yow: *Interpretable Deep Learning Model for Covid-19 Detection With Chest X-Ray Images* (2020)
16. G. Iadarola, F. Martinelli, F. Mercaldo, A. Santone : *To wards an interpretable deep learning model for mobile malware detection and family identification* (2021)
17. Cynthia Rudin, Chaofan Chen, Zhi Chen, Haiyang Huang, Lesia Semenova, Chudi Zhong : *Interpretable machine learning:Fundamental principles and 10 grand challenges* (2021)
18. Xinyang Zhang, Ningfei Wang, Hua Shen, Shouling Ji, Xiapu Luo, Ting Wang: *Interpretable Deep Learning under Fire*
19. Simon M. Thomas, James G. Lefevre , Glenn Baxter , Nicholas A. Hamilton : *Interpretable deep learning systems for multi-class segmentation and classification of non-melanoma skin cancer* (2020)
20. David S. Watson : *Conceptual challenges for interpretable machine learning* (2020)