Título do artigo (fonte 16)

Elidio da Silva, Arthur (Unipinhal) [arthur.elidio.ae@gmail.com](mailto:arthur.elidio.ae@gmail.com)

Oliveira de Freitas, Leonardo (Unipinhal) [leoff@live.com.br](mailto:leoff@live.com.br)

Resumo

Atualmente ao ir para algum posto de pronto atentimento público, pode-se notar vários casos de demora para diagnóstico, do qual a maioria são equivocados, resultando em esperas intermináveis para tratamento. Com isso em mente, o objetivo deste trabalho foi criar um sistema que fosse capaz de coletar informações de pacientes, entender o contexto dos sintomas do mesmo e com base nessas informações, ser capaz de desenvolver um diagnóstico prévio. Isso será realizado a partir dos conhecimentos obtidos em Inteligencia Artificial, Redes Neurais, Aprendizado de máquina e Aprendizado profundo com o intuito de solucionar as não conformidades apresentadas anteriormente nos locais de atendimentos de saúde, auxiliar profissionais no momento de leitura dos sintomas para o diagnóstico final, auxilar estudantes na formação acadêmica e entusiastas da área da saúde na procura de conhecimento.

Abstract

Currently in any health care station, anyone can note several cases of diagnostic delay, most of which are misleading, resulting in endless waiting for treatment. With this in mind, the objective of this work was to create a system that was able to collect information from patients, understand the context of the patient's symptoms and based on this information, be able to develop a previous diagnosis. This will be done from the knowledge obtained in Artificial Intelligence, Machine Learning and Deep Learning with the purpose of solving the nonconformities presented previously in health care station, assisting professionals at the moment of reading the symptoms for the final diagnosis, assisting students in academic training and health enthusiasts in search of knowledge.

1. Introdução

Em qualquer unidade de atendentimento de saúde nos dias de hoje, existe um processo para o atendimento do paciente a ser respeitado, inicialmente o paciente deve passar pela etapa de triagem, que consiste em um sistema de seleção, coletagem e classificação de pacientes com o intuito de definir o risco vital o paciente a partir de analise dos sintomas do mesmo. Uma vez a analise feita, o paciente é classificado de acordo com a sua respectiva urgencia vital e a partir desse momento o paciente é retornado à espera para o tratamento do doutor responsável, essa espera é respectiva a classificação atribuída, podendo chegar a até mais de 4 hrs de espera.

O objetivo inicial com o sistema desenvolvido é obter um sistema utilizando conhecimentos em Inteligencia Artificial, aprendizado de maquina, redes neurais e aprendizado profundo que, à primeiro passo, seja capaz de solucionar a demora do processo de coletagem de sintomas e classificação de pacientes, tornando assim tanto a espera para realização da triagem, a realização da triagem e a espera pós-triagem mais ágeis.

2. Fundamentos Teóricos

Para o desenvolvimento do projeto foi necessário um estudo sobre a área da saúde, mais espeficicadamente a parte de triagem, coleta de sintomas e tramento de pacientes, conceitos gerais de inteligencia artificial, aprendizado de máquina para treinamento da inteligencia artificial desenvolvida, aprendizado profundo para a inteligencia artificial se adaptar ao local designado e redes neurais para o discernimento de informação.

**Conhecimentos na área da saúde:** Após estudos em campo e conteúdos já publicados, foi obtido as sequintes informações a respeito do tratamento de pacientes em postos de pronto atendimento. A triagem tem como objetivo classificar o risco vital do paciente a partir de sintomas coletados, tais como, pressão, pulso, respiração, temperatura, glicemia capilar, peso, saturação e outros. Uma vez feito a análise de sintomas, o paciente é classificado de acordo com o seu grau de urgencia vital seguinto o padrão de classificação de manchester. De acordo com a classificação de machester, o paciente pode ser classificado em 5 tipos diferentes, são eles: Emergencia, Muita Urgencia, Urgente, Pouco Urgente e Não Urgente, dos quais são distinguidos a partir das cores Vermelho, Laranja, Amarelo, Verde e Azul respectivamente. Cada tipo de classificação contém uma previsão de atendimento, seguindo da classificação de coloração vermelha até a azul, são essas as previsões de atendimento, Imediato, em até 20 minutos, em até 60 minutos, em até 120 minutos e em até 240 minutos. A triagem tem como objetivo a coleta de informações e classificação do paciente e não o diagnóstico final do paciente.



Figura 1 – Classificação de Manchester

**Inteligencia Artificial:** O estudo em inteligencia artificial tem como objetivo replicar o comportamento humano contemplando a capacidade cognitiva, reconhecimento de contexto e tomada de decisão. Para o desenvolvimento da Inteligencia Artificial em questão foram utilizados os seguintes conhecimentos:

**Aprendizado de Máquina (Machine learning):** É a vertente da Inteligencia Artificial que utiliza dados préviamente tratatos para que o sistema seja treinado a partir do reconhecimento dos padrões de dados. No sistema em questão, utilizamos dados de um projeto que simula o estado de saúde de pacientes do estado de Massachutts conhecido como SyntheticMass.

**Redes neurais artificiais (RNA):** Redes neurais artificiais foram inspirados pelo sistema nervoso humano, em particular o cérebro, tendo como objetivo o reconhecimento de padrões para o aprendizado da Inteligencia artificial a partir de dados inputados. De forma resumida, a rede neural obtém um dado e a partir de modelos matemáticos o neuronio associa esse dado à um valor que recebe a denominação de **peso**. Esse peso é usado para a resposta final ser definida.

**Aprendizado Profundo (Deep learning):**

3. Materiais e métodos usados na implementação

Para o desenvolvimento do sistema em questão, foi utilizado a linguagem Python na IDE Spyder, usado através do programa Anaconda Navigator, que contém vários outras IDEs para desenvolvimento em Python. O Spyder foi escolhido pois disponibiliza de forma simplificada o uso das bibliotecas Keras e Numpy. NumPy é um pacote para a linguagem Python que suporta arrays e matrizes multidimensionais, possuindo uma larga coleção de funções matemáticas para trabalhar com estas estruturas. Keras é a biblioteca de código aberto e é usado para abstrair a complexidade do Tensorflow, que por sua vez, é uma biblioteca criada pelo Google com o objetivo de disponibilizar funções para facilitar o desenvolvimento de inteligências artificiais.

A arquitetura escolhida para desenvolvimento da inteligência artificial de redes neurais foi a rede de Multilayer de Perceptron, esta arquitetura faz uso do algoritmo Perceptron, que é destinado a realizar simples classificações binárias, ou seja, se pertence a uma categoria de interesse ou não. Um Perceptron é um classificador linear; ou seja, é um algoritmo que classifica a entrada separando categorias com uma linha reta. A entrada geralmente é um vetor de recursos x multiplicado por pesos w e adicionado a um viés (ou bias) b, que nada mais é do que um grau de liberdade a mais que não depende do valor de entrada passado. Um Perceptron produz uma única saída com base em várias entradas de valor real, formando uma combinação linear usando os pesos, que são calculados a partir de uma formula de ativação.

Um Multilayer Perceptron (MLP) é uma rede neural artificial composta por mais de um Perceptron. Eles são compostos por uma camada de entrada para receber o sinal, uma camada de saída que toma uma decisão ou previsão sobre a entrada, e entre esses dois, um número arbitrário de camadas ocultas que são o verdadeiro mecanismo computacional do MLP.

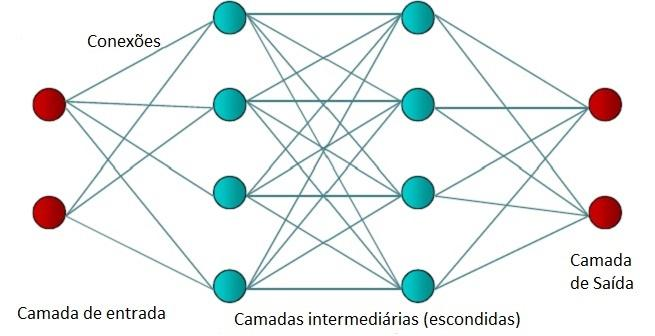


Figura 2 – Representação de uma arquitetura Multilayer Perceptron

As redes feed forward, como MLPs, são como ping-pong. Elas são principalmente envolvidas em dois movimentos, uma constante de ida e volta.  Na passagem para a frente, o fluxo de sinal se move da camada de entrada através das camadas ocultas para a camada de saída e a decisão da camada de saída é medida em relação às saídas esperadas.

Na passagem para trás, usando o backpropagation e a regra da cadeia (Chain Rule), derivadas parciais da função de erro dos vários pesos e bias são reproduzidos através do MLP. A rede continua jogando aquele jogo de ping-pong até que o erro não possa mais ser reduzido (chegou ao mínimo possível). Este estado é conhecido como convergência.

4. Resultados Obtidos

5. Conclusões

6. Referências