Universidade Tecnológica Federal do Paraná – UTFPR
Departamento Acadêmico de Computação – DACOM
Ciência da Computação
Redes Neurais Artificiais (OP63P) – C82 – 2021/1

Desenvolvimento e Treinamento de um Modelo de Rede Neural Artificial

Lucas Vinicius Ribeiro – ribeirol@alunos.utfpr.edu.br

Abril de 2021

1 Introdução

Nos últimos anos, as redes neurais artificiais (RNAs) têm sido uma das áreas da aprendizagem de máquina que mais evoluíram, tendo suas aplicações empregadas em muitos domínios diferentes. De modo geral, soluções envolvendo redes neurais artificiais são bem sucedidas para solucionar problemas com grandes volumes de dados. Alguns dos domínios em que as redes neurais artificias têm sido empregadas são: reconhecimento de imagens, reconhecimento de áudio, traduções, geração de texto, assistentes virtuais, jogos, segurança, veículos autônomos, medicina, *deep fake*, entre outros [1].

O presente experimento concentra-se em explorar o uso de redes neurais artificias em problemas de classificação de dados. Trata-se de um estudo elementar cujo objetivo é compreender o funcionamento de diferentes topologias de um modelo de RNA aplicadas em diferentes problemas. Desta maneira, pretende-se testar, para cada conjunto de dados, diferentes topologias, variando-se o número de camadas da rede e número de neurônios das camadas. Ao final, os resultados são apresentados e discutidos a fim de compreender as diferenças de acurácia para cada cenário testado.

O decorrer deste artigo está estruturado da seguinte forma: a Seção 2 apresenta os detalhes da metodologia utilizada. Os resultados e discussões são revelados na Seção 3. Por fim, são apresentadas as conclusões na Seção 4. Ao final, apresentam-se as referências bibliográficas.

2 Metodologia

O desenvolvimento deste projeto se deu em linguagem de programação Python utilizando o *framework* PyTorch. Os conjuntos de dados de treinamento foram coletados do Kaggle[1] e UCI[2]. Nesta seção, serão apresentados os detalhes a respeito da busca, coleta e normalização dos *datasets*, bem como os detalhes de implementação e execução do modelo desenvolvido.

[1] https://www.kaggle.com/datasets
[2] https://archive.ics.uci.edu/ml/index.php

Dataset	Descrição	# de Atributos	# de Instâncias	# de Classes
Balance Scale	Dataset gerado para modelar resultados experimentais psicológicos.	4	625	3
Hayes Roth	Atributos numéricos relacionados à características pessoais.	3	132	3
Mammographic Masses	Dados sobre exames de mamografia para detecção de câncer de mama.	4	961	2
Zoo	O dataset possui dados de 101 animais de um zoológico.	17	101	7
Glass	Dataset com dados para identificação de diferentes tipos de vidro.	9	214	7
Mobile Price	Dados a respeito de características presentes em celulares, para predição da faixa de preço.	20	2000	4

Tabela 1: Informações a respeito dos *datasets*.

Inicialmente, a linguagem de programação Python foi escolhida por conta da flexibilidade e recursos que tal linguagem oferece. Além disso, a implementação da rede neural artificial foi executada utilizando o PyTorch, que é um *framework* de aprendizagem de máquina, desenvolvido sob o paradigma de código aberto, que acelera o processo de desenvolvimento de modelos de aprendizagem de máquina. O PyTorch baseia-se no uso de tensores para o desenvolvimento de modelos de aprendizado profundo usando GPUs e CPUs.

Os *datasets* foram coletados das plataformas UCI e Kaggle. Ambas as plataformas possuem milhares de *datasets* disponíveis para desenvolvimento de soluções em diferentes domínios. No UCI, analisou-se a lista de *datasets*, considerando os que são destinados à criação de soluções para problemas de classificação. A lista dos *datasets* contém uma coluna indicando tal informação que é a coluna *Default Task*. Desta forma, foram selecionados três *datasets* para serem utilizados para treinamento do modelo. O mesmo processo foi aplicado no Kaggle, no entanto, foi utilizada a barra de busca para procurar por *datasets* incluindo a palavra-chave *classification*. Da mesma forma, três *datasets* foram selecionados para serem incluídos neste projeto. Ao final, seis *datasets* foram selecionados para utilizar-se no treinamento do modelo. A Tabela 2 apresenta os *datasets* e suas respectivas informações.

Após selecionados, foi realizado o procedimento de normalização dos dados, afim de padronizá-los para serem utilizados como entrada no modelo. De modo geral, a normalização se deu com base nos seguintes critérios:

- Colocar o atributo "classe"como sendo o último atributo de cada instância;
- Normalizar a "classe"em um intervalo numérico a partir de 0;
- Remover atributos não-numéricos.

Dessa forma, tendo todos os dados normalizados, a próxima etapa da metodologia foi executar o treinamento do modelo. O código-fonte está disponibilizado em `https://github.com/lucasvribeiro/neural-networks`. Basicamente, o programa possui duas funções principais: *loadDataset* e *trainModel*.

A função *loadDataset* é responsável por todo o processo de leitura e carregamento do conjunto de dados e suas respectivas classes. A função *trainModel*, por sua vez, é responsável por receber o *dataset* e criar uma rede neural artificial utilizando os recursos fornecidos pelo PyTorch. A função *run* é responsável por iterar por cada conjunto de dados e executar o treinamento do modelo. Cada época e sua respectiva perda e acurácia são impressas na saída do programa.

Sendo, o experimento foi executado da seguinte forma: para cada *dataset*, foram testadas 5 diferentes topologias, variando-se o número de camadas da rede e o número de neurônios por camada. As topologias são numeradas de 1 a 5. Em todos os *datasets* foram utilizados o mesmo conjunto de topologias. Os detalhes de cada uma delas são descritos abaixo. Vale ressaltar que, além das camadas descritas, existem mais duas camadas, sendo a camada de entrada, contendo o número de neurônios igual à quantidade de atributos do *dataset*, e a camada de saída, contendo a quantidade de neurônios referente à quantidade de possíveis classes de saída.

Topologia #01:

- Camada Oculta 1: 20 neurônios.
- Camada Oculta 2: 40 neurônios.

Topologia #02:

- Camada Oculta 1: 100 neurônios.
- Camada Oculta 2: 200 neurônios.

Topologia #03:

- Camada Oculta 1: 20 neurônios.
- Camada Oculta 2: 40 neurônios.
- Camada Oculta 3: 80 neurônios.

Topologia #04:

- Camada Oculta 1: 20 neurônios.
- Camada Oculta 2: 40 neurônios.
- Camada Oculta 3: 80 neurônios.
- Camada Oculta 4: 40 neurônios.

Topologia #05:

- Camada Oculta 1: 100 neurônios.
- Camada Oculta 2: 200 neurônios.

#	Acurácia					
Topologia	Balance Scale	Hayes Roth	Mammographic Masses	Zoo	Glass	Mobile Price
#1	0.971	0.484	0.572	0.405	0.406	0.234
#2	0.968	0.742	0.506	0.405	0.378	0.237
#3	0.912	0.356	0.605	0.405	0.341	0.234
#4	0.540	0.340	0.536	0.405	0.401	0.233
#5	0.544	0.340	0.536	0.405	0.322	0.233
Média:	0.787	0.452	0.551	0.405	0.369	0.234
Desvio Padrão:	0.224	0.172	0.038	0.000	0.036	0.001

Tabela 2: Resultados obtidos em cada experimento realizado.

- Camada Oculta 3: 250 neurônios.
- Camada Oculta 4: 150 neurônios.

Dada a quantidade de experimentos a serem realizados, o número de épocas foi definido como **100**. Além disso, a taxa de aprendizado usada foi de **0.03** e o otimizador foi o SGD. Em todas as camadas a função de ativação usada foi a Sigmoid. Os resultados obtidos e discussões são apresentados na seção a seguir.

3 Resultados e Discussões

Nesta seção serão apresentados os resultados obtidos com os testes realizados e, ao final da seção, algumas discussões são propostas a respeito dos resultados. A Tabela 3 apresenta, para cada experimento, a acurácia obtida após as 100 épocas de execução.

Dentre todos os experimentos realizados, a maior acurácia obtida foi de 0.971, utilizando a topologia com duas camadas ocultas, sendo 20 neurônios na primeira e 40 neurônios na segunda camada oculta. Neste caso, o resultado foi obtido sob o conjunto de dados *Balance Scale*, contendo 4 atributos, 625 instâncias e 3 diferentes classes possíveis. Por outro lado, a pior acurácia obtida foi de 0.233, com a execução das topologias 4 e 5 sob o conjunto de dados *Mobile Price*.

Com relação ao conjunto de dados *Balance Scale*, é possível observar que os melhores resultados foram obtidos com as topologias com menores números de camadas. Já no conjunto de dados *Mammographic Masses*, os resultados obtidos com as cinco topologias foram muito semelhantes. Além disso, a taxa média de 0.551 é uma baixa acurácia, tendo em vista que o problema possui apenas duas classificações.

É possível observar, também, que sob o conjunto de dados *Hayes Roth*, o melhor resultado obtido foi utilizando-se a topologia 2, contendo duas camadas ocultas, sendo a primeira com 100 neurônios e a segunda com 200 neurônios. Por outro lado, todas as outras topologias testadas sob este dataset resultaram em uma acurácia abaixo de 0.5.

Os experimentos executados nos *datasets Mammographic Masses*, *Zoo*, *Glass* e *Mobile Price* resultaram em desvio padrão muito baixo. Isto pode significar que o modelo ficou "preso"em um máximo local, ou também pode indicar a necessidade da realização de novos experimentos variando-se a taxa de aprendizagem, o número de neurônios por camada e também a quantidade de épocas para que o modelo possa convergir para uma boa solução.

4 Conclusões

Os experimentos realizados neste estudo contribuíram de forma significativa no que diz respeito ao aprofundamento de conceitos e fundamentos envolvendo a criação e treinamento de modelos de Redes Neurais artificiais para resolução de problemas de classificação. Em trabalhos futuros, novos experimentos variando-se taxa de aprendizagem, número de neurônios por camada, número de camadas e função de ativação, se fazem necessários para identificar o comportamento do modelo de modo a convergir em melhores resultados.

Referências

[1] Maximilian Alber, Sebastian Lapuschkin, Philipp Seegerer, Miriam Hägele, Kristof T Schütt, Grégoire Montavon, Wojciech Samek, Klaus-Robert Müller, Sven Dähne, and Pieter-Jan Kindermans. innvestigate neural networks! *J. Mach. Learn. Res.*, 20(93):1–8, 2019.