Data Science [22E3_3]

Assessment

Aluno: Frederico Flores

Também disponível em:
https://github.com/nagualcode/at-datascience-infnet

***Competência 1**: Implementar análises de dados avançadas e validar resultados*

Durante o treinamento de um algoritmo de Aprendizado por Reforço para o ambiente Lunar Lander, o seguinte dataset (link para baixar) foi gerado, com os seguintes elemento: Posição X, Posição Y, sen(theta) (theta é o ângulo de inclinação do rover em relação ao plano, ou seja, sua orientação), cos(theta), Velocidade X, Velocidade Y, Velocidade Angular (theta), Posição da Perna Esquerda, Posição da Perna Direita, Ação tomada, Recompensa, totalizando 11 entradas, sendo o último dado (Recompensa) a saída e os demais, entradas para o treinamento. Utilizando estas informação, o conjunto de dados fornecido e tendo como base o código Regressão, realize as seguintes operações:

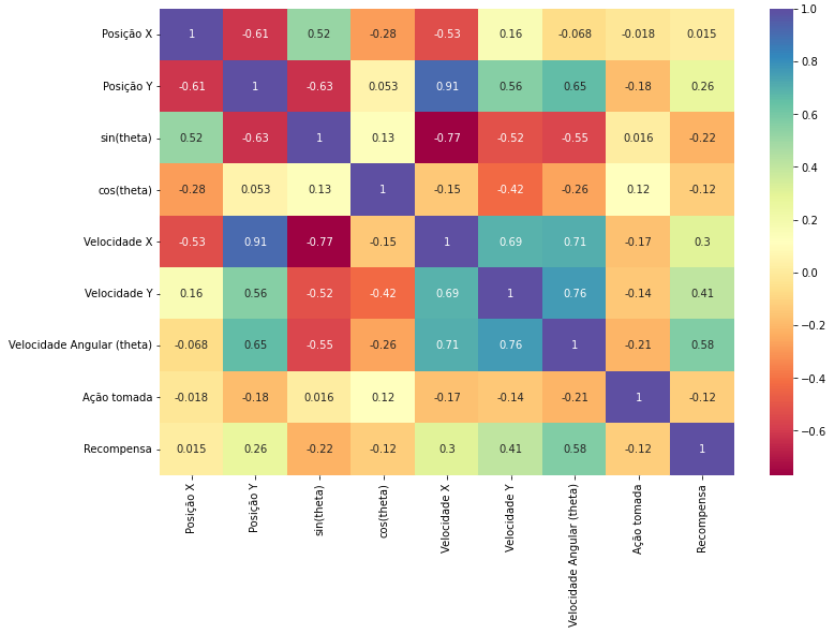
***1.1** Carregue o conjunto de dados, utilizando como nome para as colunas (column_names) o nome dos dados citados acima.*

```
columns=['Posição X',
         'Posição Y',
         'sin(theta)',
         'cos(theta)',
         'Velocidade X',
         'Velocidade Y',
         'Velocidade Angular (theta)',
         'Posição da Perna Esquerda',
         'Posição da Perna Direita',
         'Ação tomada',
         'Recompensa']

dataset = pd.read_csv('Dataset_lunar_lander_msd_1.csv',
                      header=0,
                      names=columns)

dataset.describe()
```

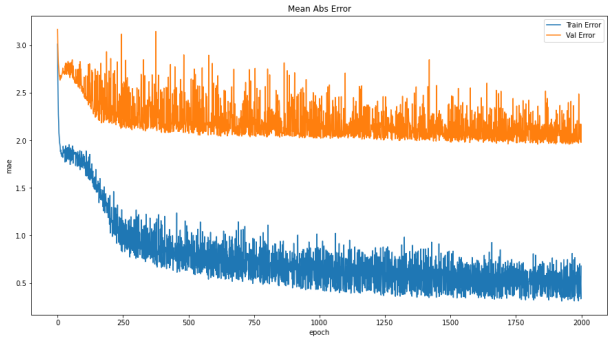
***1.2** Mostre a correlação entre a saída (Recompensa) e, pelo menos, quatro dados à sua escolha.*

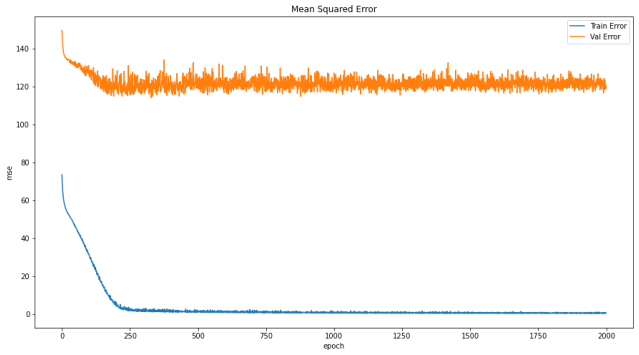


***1.3** Prepare o conjunto de dados, separando os dados de treinamento e de teste. Para este caso, não será necessário realizar a normalização dos dados.*

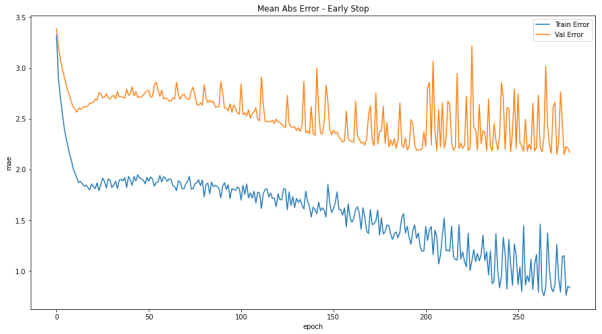
```
X = dataset.drop(columns=['Recompensa'])
y = dataset['Recompensa']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30, random_state=42)
```

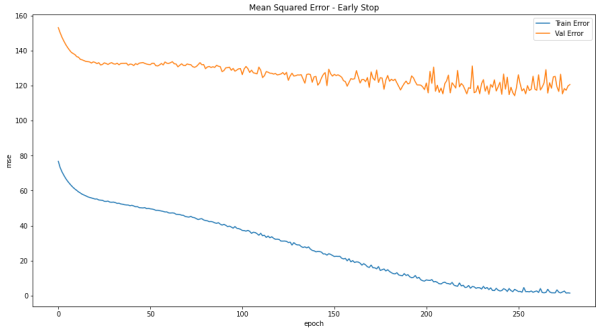
***1.4** Realize o treinamento por 2000 épocas e apresente o resultado.*





***1.5** Realizar o treinamento utilizando o método early stop e apresente o resultado.*





Respostas 1.1 a 1.5: → arquivos notebook e pdf na pasta: **[1.1-1.5]**

https://drive.google.com/file/d/1eCxTkg4-FBlZbVtHwZ46wc2QDd27W4L6/view?usp=sharing

***Competência 2**: Realizar coleta e preparação de dados para análises de dados*

***2.1** Cite três características que um conjunto de dados deve ter, discorrendo brevemente sobre cada uma delas.*

Resposta: Um bom dataset para machine learning possui três elementos chaves: **qualidade**, **quantidade** e **variabilidade**.

- **Qualidade**: Tem a ver com a riqueza e os detalhes dos dados, com relação ao objeto que estão representando. A eficiência e o tempo necessários para treinar o modelo são afetados pela qualidade do conjunto de dados que está sendo usado.

- **Quantidade**: Quanto mais quantidade, e quanto mais balanceado e distribuído entre as classes, permite um treinamento melhor, que vai resultar em mais acurácia, e evita que o modelo seja tendencioso com alguma das classes.

- **Variabilidade**: Dados que abrangem as diferentes variações e ruídos que vão. Para ajudar na variabilidade, as técnicas de data augmentation podem criar variabilidade artificial.

***2.2** A rede profunda apresentada em aula foi utilizada para realizar a classificação de imagens em duas classes (classificação binária). Com base no que foi discutido em aula, quais as modificações necessárias para que essa mesma rede seja capaz de separar os dados de entrada em mais classes?*

Resposta: No Keras a classificação multiclasses é alcançada com o softmax, que faz regressão logística multinomial. A função softmax é usada como função de ativação na camada de saída de modelos de redes neurais que predizem uma distribuição de probabilidade multinomial. Ou seja, softmax é usado como a função de ativação para problemas de classificação multiclasse em que a associação de classe é necessária em mais de dois rótulos de classe.
É comum o código que monta o modelo ter um switch case para ativar a função sigmoid no caso de 2 classes, e a softmax para mais classes, assim basta passar um argumento com o número de classes para montar o modelo correto.

***2.3** Escolha duas imagens de cada uma das duas classes, cat e dog, e faça a anotação de cada uma delas, gerando os respectivos arquivos para treinamento de rede YOLO.*

Resposta: → arquivos na pasta: **[2.3 - labelimg]**

***2.4** Utilizando o método de sua escolha, realize o processo de data augmentation para as imagens escolhidas, gerando, pelo menos, quatro variações para cada uma delas.*

Resposta: → arquivos na pasta: **[2.4 - dataaugmentation]**

***Competência 3**: Implementar análises de dados em problemas supervisionados e não-supervisionados*

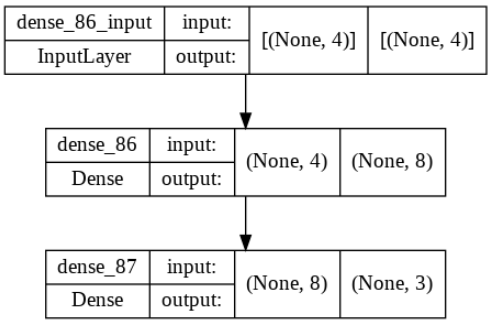
Utilizando o conjunto de dado iris, realize as seguintes operações

***3.1** Carregue o conjunto de dados, conforme mostrado em aula.*
***3.2** Utilizando os modelos estudados em aula, faça um classificador supervisionado para separar o conjunto de dados nas três categorias do conjunto de dados.*

Respostas 3.1 e 3.2: → arquivos notebook e pdf na pasta: **[3.1-3.2]**

https://colab.research.google.com/drive/1eDeMAe7axSlXugu6aMcNgKiMsHoRBoo0?usp=sharing

***3.3** Explique a arquitetura utilizada no modelo selecionado.*



Resposta: No dataset temos 4 variáveis de entrada, que são numéricas e têm a mesma escala em centímetros. Cada instância descreve as propriedades de medições de flores observadas e a variável de saída é uma espécie de íris específica.
Este é um problema de classificação multiclasse, o que significa que existem mais de duas classes a serem previstas, na verdade, existem três espécies de flores.

Os nomes das espécies/classes foram convertidos previamente para Integers com o LabelEncoder.

A topologia de rede desta rede neural simples, possui um modelo sequencial com 3 camadas em pilha, sendo: [4 inputs] -> [8 hidden nodes] -> [3 outputs].
Temos 4 dimensões de inputs (para as 4 características do dataset), com RELU, que aplica a função de ativação da unidade linear retificada. Com valores padrão, isso retorna a ativação ReLU padrão: max(x, 0) , o máximo de elemento de 0 e o tensor de entrada.

Na camada de saída, ativamos “softmax”, por ser multiclasses. Por último a rede usa o algoritmo de otimização de gradiente descendente Adam com uma função de perda logarítmica (`loss='categorical_crossentropy'`).

***3.4** Utilizando como base o algoritmo K-Mean desenvolvido em aula, realize o processo de clusterização na imagem farm_4.jpg de forma que seja possível separar as fileiras de plantas em quatro grupos.*

Resposta: → arquivos notebook e pdf na pasta: **[3.4]**

https://colab.research.google.com/drive/1qsUYAi_ewx1cAbSMbtrqgLGEwE92d0de?usp=sharing

***Competência 4**: Interpretar resultados de análises de dados avançadas*

***4.1** Utilizando os resultados obtidos durante o treinamento realizado na questão 1, faça uma comparação entre os resultados obtidos durante o treinamento inicial e o que utilizou early stop.*

Resposta: O treino com o early stop conseguiu ser o suficiente para treinar o modelo de maneira eficiente, parando no momento que o erro médio atingiu o seu mínimo (por volta da época 250), e assim evitando o overfitting.

***4.2** Para o modelo treinado na questão 1, sem early stop, faça uma comparação entre o valor real que consta na coluna reward (label) e os resultados obtidos para a inferência de um conjunto de exemplos antes e depois do treinamento do modelo.*

Resposta: Podemos verificar que em ambos os casos o valor ideal de recompensas (1) aconteceu por volta da época 250, e se estabilizou desde então. Indicando que o treinamento está balanceado e ser overfitting.

***4.3** Compare o resultado obtido com o método utilizado na questão 3.1 para classificação com o método K-Means usado em aula para a separação do mesmo conjunto de dados, apresentando suas respectivas vantagens e desvantagens.*

Resposta: O método supervisionado que utilizei na questão 3, Keras, foi capaz de treinar com a etiquetagem de 20% do dataset, e classificar com sucesso os outros 70%, estabilizando a acurácia em 96% por volta do época 40. Porém o K-means usado em aula mostrou eficiencia de 100%, com a vantagem de não precisar da etiquetagem.

***4.4** Qual a vantagem de se utilizar um método de clusterização como o K-Means em uma aplicação que tem como entrada o tipo de dado usado na questão 3 (imagens).*

Resposta: K-means certamente é melhor para encontrar padrões e classificar em clusters, devido a sua eficiência e praticidade, dispensando a supervisão (etiquetagem), além de escalona para grandes conjuntos de dados.