



Inteligência Artificial Computacional

Trabalho Computacional AV3: Redes Neurais Artificiais.

Professor: Prof. Msc. Paulo Cirillo Souza Barbosa

Observações iniciais.

Solicita-se que realize a leitura por completo do presente documento. Na AV3 as bibliotecas permitidas são:

1. numpy;
2. matplotlib;
3. seaborn;
4. opencv (esta última apenas para realizar a leitura de imagens e o redimensionamento)
5. os;

Introdução.

O presente trabalho é composto por duas etapas em que deve-se utilizar os modelos de IA, que são bioinspirados no funcionamento do cérebro humano. Tais modelos são redes neurais artificiais (RNA) lineares e não-lineares. A primeira etapa do trabalho, trata-se da solução de um problema simples, e serve como um validador das implementações dos modelos solicitados. A segunda etapa, trata-se da resolução de um problema de reconhecimento facial de imagens.

Primeira etapa [3,0 pts]

Solicita-se que faça o acesso ao conjunto de dados disponibilizado na plataforma AVA, chamado **spiral.csv** que trata-se de dados sintetizados que possuem duas classes (as quais estão rotuladas no próprio arquivo na terceira coluna).

1. Faça uma visualização inicial dos dados através do gráfico de espalhamento.
2. O presente conjunto de dados possui $p = 2$, $N = 2000$ e $C = 2$. Assim, faça a organização do conjunto de dados para serem apresentados às redes neurais.
3. Os modelos de RNA a serem implementados nessa etapa serão: **Perceptron Simples**, **ADaptive LINear Element**, **Perceptron de Múltiplas Camadas (MLP)**. Para cada modelo, deve-se discutir como os hiperparâmetros foram escolhidos. **Obs:** No final do presente documento, existem informações adicionais sobre tais modelos.
4. Para validar os modelos utilizados na tarefa, sua equipe deve projetar a validação utilizando as simulações por Monte Carlo. Nessa etapa, defina a quantidade de rodadas da simulação igual a $R = 500$. Em cada rodada, deve-se realizar o particionamento em 80% dos dados para treinamento e 20% para teste. As medidas de desempenho de cada modelo, devem ser: **acurácia**, **sensibilidade** e **especificidade**.
5. Das 500 rodadas do item anterior, selecione aquelas que proporcionaram o maior e o menor valor de acurácia. Para esses dois casos, construa a matriz de confusão. **Obs:** sua matriz de confusão deve ser implementada pela equipe, no entanto, para realizar seu plot, pode-se utilizar a biblioteca *seaborn*. Exemplo: [link heatmap](#). Além disso, para os dois casos (maior e menor acurácia), pede-se que plote a curva de aprendizado de todos os modelos.

6. Ao final das $R = 500$ rodadas calcule para cada modelo utilizado, média aritmética, desvio-padrão, valor maior, valor menor, para as seguintes métricas de desempenho: **acurácia**, **sensibilidade** e **especificidade**. Coloque os resultados obtidos em uma tabela (UMA PARA CADA MÉTRICA) e **discuta os resultados obtidos**. **Obs:** O resultado não precisa ser limitado a tabela, como pode ser expresso via gráficos (BOXPLOT, VIOLIN PLOT E OUTROS). Como exemplo, segue uma tabela que serve para uma das três métricas solicitadas.
7. Para o modelo MLP, pede-se que identifique o *underfitting* e *overfitting*. Neste caso, sua equipe deve explorar diferentes topologias ao modificar os hiperparâmetros associados a quantidade de neurônios e camadas (realizando um subdimensionamento e superdimensionamento). Para os dois casos, faça a composição do resultado ao descrever a topologia encontrada e meça o desempenho utilizando matriz de confusão, acurácia, especificidade, sensibilidade e curva de aprendizado.

Modelos	Média	Desvio-Padrão	Maior Valor	Menor Valor
Perceptron Simples.				
ADaptive LINear Element.				
Perceptron de Múltiplas Camadas				

Reconhecimento Facial[7,0 pts]

A segunda etapa do trabalho a tarefa classificação de pessoas a partir de imagens de faces, sendo necessário que seu programa lide com imagens. Para isso, faça a instalação do **opencv** utilizando o seguinte comando no terminal:

- `pip install opencv-python`

Com esta biblioteca, é possível fazer as aquisições e transformações necessárias de modo que as imagens possam ser utilizadas nos algoritmos de aprendizado de máquina vistos em sala de aula. Após utilizar o **import cv2**, os comandos a serem utilizados são:

- `cv2.imread(path, cv2.IMREAD_GRAYSCALE)` para realizar a leitura de uma imagem já no formato em escala de cinza no qual o caminho está em **path**;
- `cv2.resize(imagem, (dim1, dim2))` que realiza o redimensionamento de uma **imagem** para **dim1** × **dim2** *pixels*.

O conjunto de dados está disponibilizado no AVA e consiste em 640 imagem em escala de cinza de 20 pessoas diferentes, com poses diferentes (straight, left, right, up), com expressões diferentes (neutral, happy, sad, angry) e usando ou não óculos. No entanto, pede-se que resolva a tarefa de reconhecimento facial. Como informação adicional, afirma-se que na pasta chamada RecFac, encontram-se 20 outros diretórios referentes a cada um dos indivíduos, e em cada pasta desta, há um total de 32 imagens com as configurações explicitadas anteriormente.

Após essa etapa inicial, faça o que se pede:

1. Utilizando o comando **resize** descrito anteriormente, faça uma experimentação de diferentes tamanhos, por exemplo, 50×50 , 40×40 , 30×30 , 20×20 , 10×10 . Escolha aquele que levar ao melhor compromisso (*tradeoff*) entre acurácia/sensibilidade/especificidade e custo computacional.
2. Organize as imagens de modo que se tenha, as matrizes

$$\mathbf{X} \in \mathbb{R}^{p+1 \times N} \quad \mathbf{Y} \in \mathbb{R}^{C \times N}$$

Neste caso, leve em consideração a codificação *one-hot-encoding*

3. Os modelos de RNA a serem implementados nessa etapa serão: **Perceptron Simples**, **ADaptive LINear Element**, **Perceptron de Múltiplas Camadas (MLP)**. Para cada modelo, deve-se discutir como os hiperparâmetros foram escolhidos. **Obs:** No final do presente documento, existem informações adicionais sobre tais modelos.

4. Para validar os modelos utilizados na tarefa, sua equipe deve projetar a validação utilizando as simulações por Monte Carlo. Nessa etapa, defina a quantidade de rodadas da simulação igual a $R = 50$. Em cada rodada, deve-se realizar o particionamento em 80% dos dados para treinamento e 20% para teste. As medidas de desempenho de cada modelo, devem ser: **acurácia**, **sensibilidade** e **especificidade**.
5. Das 50 rodadas do item anterior, selecione aquelas que proporcionaram o maior e o menor valor de acurácia. Para esses dois casos, construa a matriz de confusão. **Obs:** sua matriz de confusão deve ser implementada pela equipe, no entanto, para realizar seu plot, pode-se utilizar a biblioteca *seaborn*. Exemplo: [link heatmap](#). Com essas matrizes de confusão criadas, tente entender para qual categoria os modelos fazem previsões piores e melhores. Tente verificar se há um padrão percebido nos modelos em questão (sempre erram ou sempre acertam as mesmas categorias). Além disso, para os dois casos (maior e menor acurácia), pede-se que plote a curva de aprendizado de todos os modelos.
6. Ao final das $R = 50$ rodadas, calcule para cada modelo utilizado, média aritmética, desvio-padrão, valor maior, valor menor, para as seguintes métricas de desempenho: **acurácia**. Coloque os resultados obtidos em uma tabela **ediscuta os resultados obtidos**. **Obs:** O resultado não precisa ser limitado a tabela, como pode ser expresso via gráficos (BOXPLOT, VIOLIN PLOT E OUTROS). Como exemplo, segue uma tabela que serve para uma das três métricas solicitadas.

Modelos	Média	Desvio-Padrão	Maior Valor	Menor Valor
Perceptron Simples.				
ADaptive LINEar Element.				
Perceptron de Múltiplas Camadas				

Informações Adicionais aos modelos de RNA

De maneira geral, para as duas etapas do trabalho, sua equipe deve considerar que tais modelos possuem hiperparâmetros e procedimentos específicos. Para tais modelos, é necessário que os dados sejam normalizados (caso eles não já estejam). Além disso, todos os modelos precisam de um critério de convergência baseado no número máximo de épocas. Essa última informação é importante, pois, o modelo Perceptron Simples tradicional não possui esse critério. Assim, tal modificação deve ser feita pela sua equipe.

Considere que para os modelos Perceptron Simples e ADALINE, a função de ativação é linear (degrau ou degrau bipolar). A função de ativação utilizada na rede MLP deve ser escolhida entre a tangente hiperbólica e a sigmoide logística. Para simplicidade do projeto, faça com que todos os neurônios possuam a mesma função.

Para o caso da rede MLP, considere escolha dos hiperparâmetros, conforme as discussões realizadas em sala (ou através do que está descrito nos slides). Os hiperparâmetros são:

1. Quantidade de camadas escondidas.
2. Quantidade de neurônios nas camadas escondidas.
3. Quantidade de neurônios na camada de saída.
4. Valor da precisão (ou critério de parada da convergência).
5. Número máximo de épocas.
6. Taxa de aprendizagem.

5) Relatório.

Além das implementações, o presente trabalho deve ser entregue em modelo de relatório. Este deve possuir as características descritas nos slides de apresentação do curso. Desta maneira, deve possuir:

1. Título (2,5%).
2. Resumo (2,5%).
3. Metodologia (42,5%).

4. Resultados (42,5%).

5. Conclusões (10%).

O modelo para trabalho pode ser encontrado neste [LINK](#)

6) Observações.

- **Obs1:** O envio das implementações é **obrigatório**. Caso a equipe não realize esta entrega, será atribuído nota **zero** para os respectivos alunos.
- **Obs2:** A data estipulada para entrega do trabalho, também é um critério avaliativo. Assim, caso haja atraso na entrega do trabalho, será aplicada: **de 00:15h até 24h: penalidade de 20% ; 24:15h até 48h: penalidade de 40% ; acima de 48h: penalização máxima (100%)**.
- **Obs 3:** As implementações devem ser realizadas utilizando apenas as bibliotecas descritas no início do presente documento. Caso alguma biblioteca diferente seja utilizada, será atribuída a nota **zero** para todos os membros da equipe.
- **Obs 4:** QUAISQUER alterações dos modelos, que não foram discutidas em sala de aula, ou que não estão presentes no conjunto de slides, devem ser explicadas pela equipe. Caso a equipe não saiba o motivo pelo qual as adequações foram realizadas, será atribuída a nota **zero** aos respectivos alunos.
- Caso não sejam utilizados os modelos descritos no presente documento, será considerado que sua equipe não atendeu aos requisitos, e assim, a nota para os membros da equipe será **zero**.
- **Obs 5:** A apresentação do trabalho no formato de arguição, é obrigatória. A data da apresentação está definida no AVA. Se a equipe não participar deste momento, a nota do trabalho será ZERO.
- **Obs 6:** Os trabalhos e implementações serão enviadas a um software anti-plágio. Qualquer caracterização de plágio ocasionará em nota zero para ambas equipes.