13 - Prática: Redes Neurais (II)

Descrição da Atividade

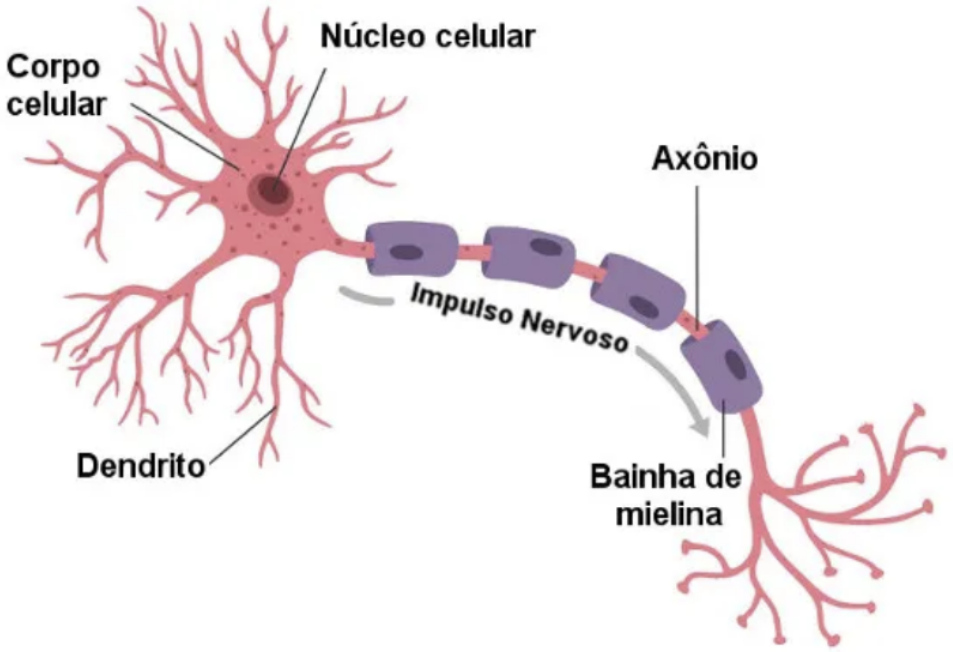
Seção 2: Redes Neurais Artificiais

Aprendizagem supervisionada	Aprendizagem não supervisionada
Redes Neurais Artificiais classificação e regressão	Mapas auto organizáveis detecção de características e agrupamento
Redes Neurais Convolucionais visão computacional	Boltzmann machines sistemas de recomendação redução de dimensionalidade
Redes Neurais Recorrentes análise de séries temporais	Autoencoders redução de dimensionalidade
	Redes adversariais generativas geração de imagens

Seção 3: Teoria resumida sobre redes neurais artificiais.

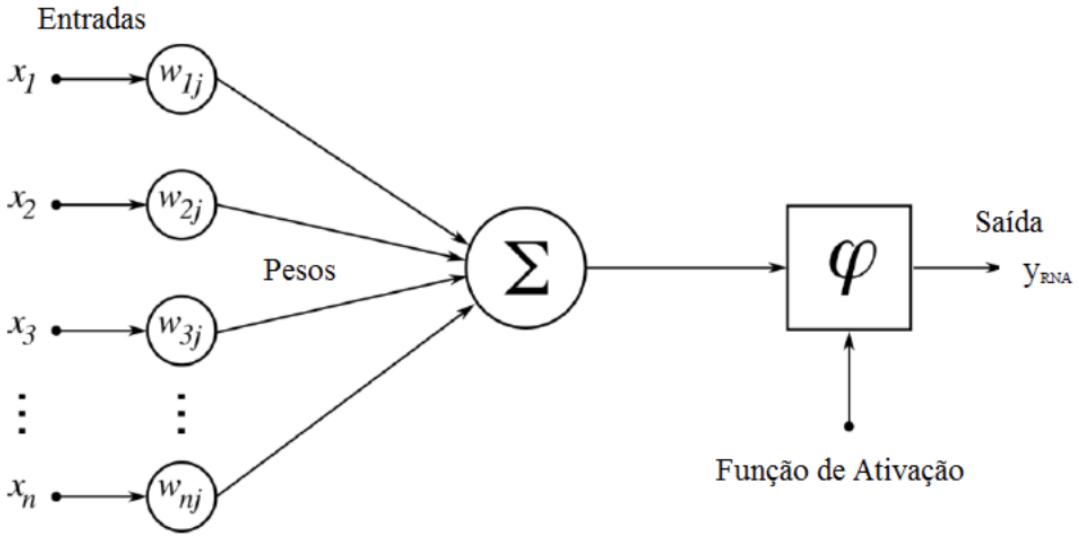
1.Fundamentos biológicos

- Redes Neurais e Neurônios
  - A troca de informação entre os neurônios é o que permite você a executar determinadas atividades
  - Neurônio: Dendritos, Corpo Celular, Axônio e terminais do axônio
  - Sinapse: troca de informações



2.Perceptron de uma camada

- Neurônio Artificial
  - FunçãoSoma = (Entrada1 * Peso1) + (Entrada2 * Peso2)...
  - StepFunction = FuncaoSoma maior que 1, entao saida = 1. Senão = 0.



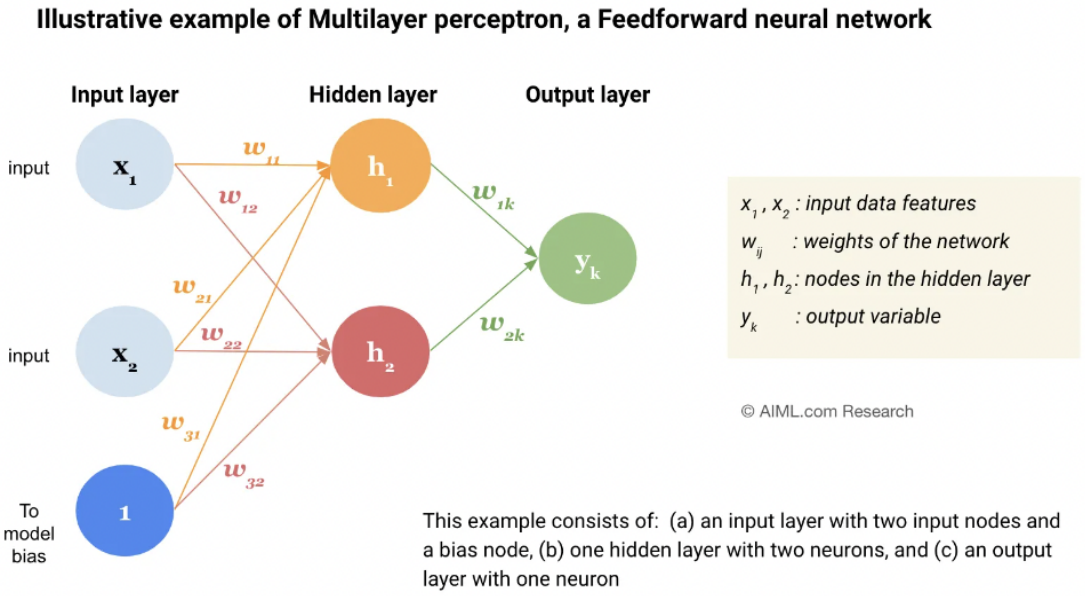
  - Utilizamos o operador ‘e’ para prever a saida.
    - Outros Operadores como o ‘Xor’
    - Operador ‘Ou’

- Vemos quantos ela acertou e quanto errou.
- Arrumamos onde ela errou. Peso(n+1) = peso(n) + (TaxaAprendizagem*entrada*erro)

3.Redes multicamada - função soma e função de ativação

- Existem várias funções de Ativação, uma das mais comuns é a Sigmoide. Y = 1/1+e-x

1. Pesos colocados aleatoriamente pela biblioteca
2. Função soma na camada escondida
3. Com os resultados da função soma na cama escondida é aplicada a função de ativação, neste caso a Sigmoide
4. Saida com os resultados
5. Aplicação da função soma em cima dos resultados



4.Redes multicamada - cálculo do erro

- Erro = respostaCorreta - respostaCalculada
- Media Absoluta = faz a media do erro, o objetivo é sempre diminuir.
- Epocas = Define quantas épocas e vai alterando para modificar os pesos

5.Descida do gradiente - arrumar o erro

- Tendo o resultado da funcao de ativação (Sigmoide) - Aplicamos na função da derivada = D = Y*(1-Y)

6.Cálculo do parâmetro delta

1. Função Ativação
2. Derivada da Função
3. Delta
4. Gradiente

7.Ajuste dos pesos com backpropagation

- PesoN+1 = (pesoN*momento)+(entrada * delta * taxa de aprendizagem)
- Faz o cálculo de esquerda para direita e de direita para esquerda
- Faz os calculos camada por camada do neuronio oculto
- Neste exemplo:
  - Taxa de Aprendizagem = 0.3
  - Momento = 1
  - Entrada X Delta = Delta 1 0.032; Delta 2 0.022; Delta 3 0.021
  - Dado esses valores, agora começa o cálculo da esquerda para direita.
    Da Saída para a Camada Oculta
    1. Camada Oculta = PesoN+1 = (pesoN*momento)+(entrada * delta * taxa de aprendizagem) = (-0.017∗1) + 0.032∗0.3 = -0.007
    2. Camada Oculta= PesoN+1 = (pesoN*momento)+(entrada * delta * taxa de aprendizagem) = (-0.893∗1) + 0.022∗0.3 = -0.886
    3. Camada Oculta = PesoN+1 = (pesoN*momento)+(entrada * delta * taxa de aprendizagem) = (0.148∗1) + 0.021∗0.3 = -0.154
  - **Da camada oculta para a camada de entrada.**
    1. Camada Oculta
       a. Delta Camada Oculta * Primeiro Valor de Entrada = **valor1** 0∗0.000 + **valor2** 0*(-0.001) + **valor3** 1*(-0.001)+ **valor4** 1∗0.000 = -0.000 arredondado

b. Delta Camada Oculta * Segundo Valor de Entrada = **valor1** 0∗0.000 + **valor2** 1*(-0.001) + **valor3** 0*(-0.001)+ **valor4** 1∗0.000 = -0.000 arredondado

2. Camada Oculta

a. Delta Camada Oculta * Primeiro Valor de Entrada = **valor1** 0∗0.022 + **valor2** 0*(-0.029) + **valor3** 1*(-0.027)+ **valor4** 1∗0.017= **-0.010** arredondado

b. Delta Camada Oculta * Segundo Valor de Entrada = **valor1** 0∗0.022 + **valor2** 1*(-0.029) + **valor3** 1*(-0.027)+ **valor4** 1∗0.017 = -**0.012** arredondado

3. Camada Oculta

a. Delta Camada Oculta * Primeiro Valor de Entrada = **valor1** 0* (-0.004) + **valor2** 0∗0.005 + **valor3** 1*(-0.004)+ **valor4** 1* (-0.003) = **0.001** arredondado

b. Delta Camada Oculta * Segundo Valor de Entrada = **valor1** 0* (-0.004) + **valor2** 0∗0.005 + **valor3** 1∗0.004+ **valor4** 1*(-0.003) = **0.002** arredondado

- Taxa de Aprendizagem = 0.3
- Momento = 1
- Entrada x Deltra = -0.000; -0.010; 0.001; -0.000; -0.012; 0.002
- (-0.424∗1) + (-0.000) * 0.3 = -0.424
- (0.358∗1) + (-0.000) * 0.3 = 0.358
- (-0.740∗1) + (-0.010) * 0.3 = -0.743
- (-0.577∗1) + (-0.012) * 0.3 = —0.581
- (-0.961∗1) + (0.001) * 0.3 = -0.961
- (-0.469∗1) + (0.002) * 0.3 = -0.468

- ÉPOCA = Número de vezes que esse ajuste de pesos será executado. Neste caso definimos como 1 época

8.Bias, erro, descida do gradiente estocástico e mais parâmetros

- Bias = Adiciona uma camada ficticia para melhorar os resultados da camada neural.
    - Valores diferentes mesmo se todas as entradas forem zero
    - Muda a saída com a unidade de bias
    - Maior parte das bibliotecas é adicionado automaticamente
- Erro
    - Algoritmo mais simples. Erro = RespostaCorreta - RespostaCalculada
    - MSE (Mean Squared Error)e (RMSE)Root Mean Squared Error. Erros maiores são penalizados.
        1. (0-0.406)ˆ2 = 0.164.
        2. MSE = Soma/ qtde = 1.011/4 = 0.252
        3. RMSE = Raiz0.252 = 0.501
- Batch Gradient Descent
    - Calcula o erro para todos os registros e atualiza os pesos
- Stochastic Gradient Descent
    - Calcula o erro para cada registro e atualiza os pesos
    - Mais rápido - não precisa carregar todos os dados em memoria
- Mini Batch Gradient Descent
    - Escolhe um número de registros para rodar e atualizar os pesos. Lote em Lote.
- Parametros
    - Learning Rate (Taxa de Aprendizagem) - Quao rápido o algoritmo vai fazer as atualizações dos pesos
    - Batch size (tamanho do lote)
    - Epochs (épocas)
- Funções de Ativação
    - Step (Degrau) - Valor 0 ou 1
    - Sigmoide - Valores entre 0 e 1
    - Tangente Hiperbólica - Valores entre -1 e 1

- Relu (Rectified Linear Units) - Valores≥0 (sem valor máximo)
- Softmax - problemas com mais de 3 classes. Exemplo: detectar bolas, cachorros, maquinas... vai retornar a probabilidade
- Linear - valores positivos e negativas, problemas de regressão. Exemplo: gastos do cartão de crédito

9.Funções de ativação - implementação I

- Atividade no Notebook

10.Funções de ativação - implementação II

- Atividade no Notebook

Seção 4: Classificação binária - base breast cancer

1. Base de dados breast cancer

- Base de dados online - https://archive.ics.uci.edu/
- Preparação dos dados

2. Estrutura da rede neural

- 30 Neuronios = 30 paramentros
- Camada Oculta Densa = Todos os neuronios estao conectados a todas as camadas
- 16 na camada oculta, seguindo a formula 30+1(saida binaria)/2 = 15.5 arrendodamos para 16
- Funcao RELU na primeira camada
- Funcao Sigmoide na Camada de saida para me dar a probabilidade

3. Configuração e execução da rede neural

- Atividade no Notebook

4. Previsões com a rede neural

- Atividade no Notebook

5. Mais camadas e parâmetros do otimizador

- Adicionamos mais uma camada = e nesse caso ela deixou o modelo com menos acuracia, como é uma base de dados simples, nao precisa necessariamente precisa de tantas camadas
- O ideal é fazer varios testes, adicionado e retirando camada até chegar na arquitetura ideal, que de mais acuracia para o modelo

6. Visualização dos pesos

- Após executar o modelo é possível verificar os pesos utilizados

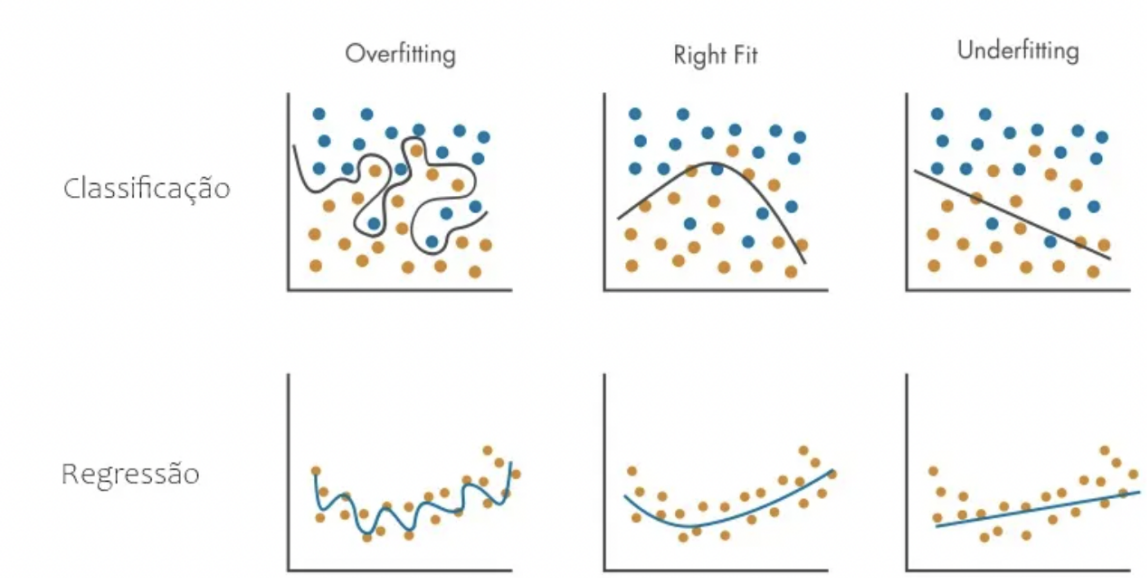
7. Validação cruzada - teoria (K-Fold Cross Validation)

- Serve para dividir a base de dados de forma mais eficiente, entre o que vai ser usado para Treinar x Testar
- Utilizamos um valor K. Por exemplo K = 4
- A base é dividida em 4

8. Validação cruzada - implementação

9. Overfitting e underfitting - teoria

- Overfitting: A linha se adaptou demais aos dados, ou seja na base de teste vai ter resultados ótimos, mas com outros dados não vai se adaptar.
- Undefitting: Algoritmo muito simples pra resolver um problema complexo. A linha não passa perto dos pontos.



10. Overfitting e dropout - implementação

11. Tuning (ajuste) dos parâmetros - implementação

12. Classificação de somente um registro - implementação

13. Salvar e carregar a rede neural - implementação

14. Tarefa 1: Melhoria dos resultados na base breast cancer - implementação

Seção 5: Classificação multiclasse - base iris

1. Base de dados iris - prática

2. Estrutura da rede neural

- Problema de classificação de mais de 2 classes, utilizamos neste caso a função Softmax

3. Previsões com a rede neural

- Utilizou a Função Sigmoid para validar, ou seja >0.5 = False

4. Validação cruzada

5. Tarefa 2: Tuning dos parâmetros

6. Tarefa 3: Salvar o classificador e classificar somente uma planta

Seção 6: Regressão - base de carros usados

1. Base de dados de carros usados

- Utilizando regressão para prever o valor do carro.

2. Pré-processamento - valores inconsistentes

- Tratamento dos dados inconsistentes

3. Pré-processamento - valores faltantes

- Tratamento dos dados faltantes

4. Pré-processamento - one hot encoder

- Todas as categorias sendo transformadas em 0 e 1. Se por exemplo eu tiver 3 categorias, a matrix fica assim:

Color
Red
Green
Blue

→

Color	Red	Green	Blue
Red	1	0	0
Green	0	1	0
Blue	0	0	1

5. Estrutura da rede neural

- Por enquanto se a aplicacao do Dropout

6. Validação cruzada

Seção 7: Regressão com múltiplas saídas - base vídeo games

1. Base de dados vídeo games

- Previsão de Vendas em várias regiões

2. Pré-processamento

3. Estrutura da rede neural

- Erros maiores são penalizados maior. Por isso nesse caso usamos o MSE (Mean Squared Error).

Conclusão

O conteúdo apresenta uma visão bem estruturada sobre o funcionamento das redes neurais artificiais, desde seus fundamentos biológicos até exemplos práticos. Explica como os neurônios biológicos inspiraram o desenvolvimento de neurônios artificiais. Aprofunda na construção de redes neurais, mostrando conceitos de perceptrons de camada única, redes multicamada, cálculos de erro e ajustes de pesos utilizando o backpropagation.

Também aborda, diferentes métodos para otimização, como: descida do gradiente e suas variantes (estocástica e mini-batch), learning rate, batch size e epochs. As funções de ativação são apresentadas nos exemplos práticos, exemplifiando sua importância para tarefas específicas, como classificação e regressão.

Referências

Artigo disponível em <https://www.researchgate.net/figure/Figura-1-Representacao-do-neuronio-artificial_fig1_329245206>.

Artigo disponível em <https://blog.betrybe.com/tecnologia/operadores-logicos/>.

Artigo disponível em <https://aiml.com/what-is-a-multilayer-perceptron-mlp/>.

Artigo disponível em <https://aiml.com/what-is-a-multilayer-perceptron-mlp/>.

Artigo disponível em <https://www.datageeks.com.br/overfitting/>.

Artigo disponível em <https://www.datacamp.com/pt/tutorial/one-hot-encoding-python-tutorial>.