

Project Pegasus: Identificando perfis de reprovação na disciplina de Lógica de Programação usando os Mapas Auto-Organizáveis de Kohonen (SOM)

Introdução

Este experimento foi desenvolvido por Vilson Rodrigues Câmara Neto. Um problema recorrente na Universidade Federal do Rio Grande do Norte (UFRN) é a dificuldade dos estudantes em compreender lógica de programação. Visto tal problemática, foi realizado este experimento com os estudantes da Escola de Ciências e Tecnologia (ECT) da UFRN, de forma tentar identificar padrões de reprovação de alunos que já cursaram a disciplina em anos anteriores e usaram a plataforma LoP. Foi realizado um tratamento dos dados para unir as tabelas do LoP com as submissões por semana e a sua situação final. Com essa base de dados foi escolhido o algoritmo de redes neurais, Mapas Auto-Organizáveis.

Proposto por Teuvo Kohonen em 1982, a rede neural artificial (RNA) de aprendizado não-supervisionado, SOM, busca diminuir a dimensionalidade de um grupo de dados mas mantendo as relações reais, e ao fim, um conjunto de multidimensões vira um conjunto bidimensional, adicionando valores de forma que os similares fiquem próximos. Diferente de outras RNA's que usam aprendizagem de correção de erros, a rede SOM utiliza aprendizagem competitiva. Os princípios do aprendizado competitivo são:

- Um conjunto de neurônios que são todos iguais, exceto por alguns pesos sinápticos distribuídos aleatoriamente e que, portanto, respondem diferentemente a um determinado conjunto de padrões de entrada;
- Um limite imposto à "força" de cada neurônio;
- Um mecanismo que permite aos neurônios competir pelo direito de responder a um determinado subconjunto de entradas, de modo que apenas um neurônio de saída (ou apenas um neurônio por grupo) esteja

ativo (por exemplo, "ativado") de cada vez. O neurônio que vence a competição é chamado de neurônio "vencedor leva tudo" .

Os neurônios individuais da rede aprendem a se especializar em conjuntos de padrões semelhantes e, assim, tornam-se "detectores de recursos" para diferentes classes de padrões de entrada. Quando um exemplo é apresentado à rede, sua distância euclidiana para os vetores de peso é calculada. O neurônio no qual o vetor de peso é mais semelhante à entrada é chamado de melhor unidade correspondente ou vencedor. Os pesos do vencedor e dos neurônios próximos a ele são ajustados para o vetor de entrada. A taxa de aprendizagem da rede diminui ao passar das iterações/épocas. A fórmula de atualização dos pesos para um neurônio v com vetor de peso $W_v(s)$ é:

$$W_v(s+1) = W_v(s) + \theta(u, v, s) \cdot \alpha(s) \cdot (D(t) - W_v(s))$$

Figura 1: Fórmula de atualização dos pesos dos Mapas Auto-Organizáveis

sendo s o índice das etapas, t o índice na amostra de treinamento, u é o índice do vencedor para a entrada D , $\alpha(s)$ é a taxa de aprendizado que diminui com o tempo e $U(u, v, s)$ é a função de vizinhança que fornece a distância entre o neurônio u e o neurônio v na etapa s .

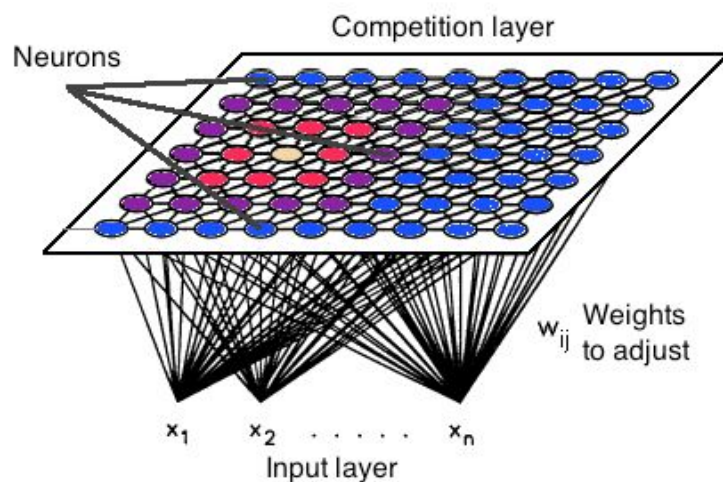


Figura 2: Representação do funcionamento dos Mapas Auto-Organizáveis

Metodologia

A base de dados contém as submissões de questões durante 18 semanas e a situação final dos alunos que cursaram a disciplina em semestres anteriores, ficou dividido como: 0 para reprovados e 1 para aprovados.

O motivo de utilizar as submissões é poder visualizar como as interação do aluno como a plataforma LoP influenciam no seu resultado final na disciplina.

Como é uma RNA de aprendizado não-supervisionado, então existe apenas o processo de treinamento. Foram utilizados 948 ocorrências, que corresponde a toda a base de dados, com os alunos que cursaram entre 2017.2 e 2019.1.

Foi aplicada uma normalização MinMaxScaler a fim de evitar que pesos altos influenciem seus adjacentes, deixando os valores entre 0 e 1.

O objetivo é tentar visualizar através de grafos, como o aluno está em relação aos semestres anteriores nas mesmas condições. Para isso foram definidos 3 critérios para visualização: até 10% de reprovados no neurônio vencedor do estudante a cor escolhida para o nó do grafo é verde, ou seja, uma região segura com base no passado; entre 10% e 20% de reprovados, o nó é amarelo; e acima de 20% de reprovados, o estudante estaria em uma situação perigosa, o nó terá cor vermelha.

Outras formas de visualização apresentadas são: exibição dos pesos de um neurônio escolhido, gráfico de superfície e gráfico de pizza.

Códigos

```
tamanhoXdaRede = 13
tamanhoYdaRede = 13
quantidadeCaracteristicas = 18

from minisom import MiniSom
som = MiniSom(x = tamanhoXdaRede,
              y = tamanhoYdaRede,
              input_len = quantidadeCaracteristicas,
              sigma = 1.0,
              learning_rate = 0.5)
```

Figura 3: Modelo da rede SOM

```
som.train_random(data = X_train,
                 num_iteration = 100000,
                 verbose = True)
```

Figura 4: Parâmetros de treinamento

```
# matriz de zeros para contador de reprovados
MContRe = np.zeros((tamanhoXdaRede,tamanhoYdaRede))
# matriz de zeros para contador de aprovados
MContAp = np.zeros((tamanhoXdaRede,tamanhoYdaRede))
# matriz de zeros para o número total de alunos
MContT = np.zeros((tamanhoXdaRede,tamanhoYdaRede))

cont = 0;

for x in X_train:
    pos = som.winner(x)
    if (Y_train[cont] < 5): #Reprovado
        MContRe[pos] += 1
    if (Y_train[cont] >= 5): #Aprovado
        MContAp[pos] += 1
    MContT[pos] += 1
    cont= cont+1
```

Figura 5: Gerador das matrizes

Como temos o output (a nota) dos estudantes dos semestre anteriores, podemos marcar em uma coordenada numa matriz de zeros se a condição for satisfeita, que nesse caso, foram de aprovação para nota final maior ou igual a 5, e reprovação para menor que 5, também foi computada todas as ocorrências.

Resultados

A rede treinada gerou o mapa visto nas figuras a seguir, uma tendo todas as ocorrências e outra apenas dos situação de reprovados. Vemos que a SOM foi eficiente em reduzir a dimensionalidade do problema e transformar em um problema bi dimensional, mantendo características do problema inicial.

```
print(MContT.astype(int))
```

```
[[ 118  14 245 197  54  16 5467  0 387 405 104 100 36]
 [  23  78  42  16 246 180  92 184 211  31  64  99 33]
 [ 123  56   8 129  65  34 105 151 282 128 49 76 30]
 [  98  77 409  14  96 221 241 111 213 108 117 93 43]
 [ 115 146  45 211  54 189  36  65 148  77  85 92 48]
 [  65  98  95 244 115  16  19 137  57  39  32 66 32]
 [ 121  47  81 103 101  74  68  48  54  81  13 27 65]
 [  35  57  62  90  78  31  53  30  65  56  77 58 26]
 [  29  90  56  80  63  84  26  74  6 113  68 43 40]
 [  73  21  13  57  69  44  62  21  55  38 127 38 48]
 [  36  29  59  47  48  61  70 104  59  69  39 75 69]
 [  12  66   0  44  44  57  88  54  72  47  69  6 65]
 [  75  94  25  40  65  34  22  62  41  45  62 57 26]]
```

```
print(MContRe.astype(int))
```

```
[[  5  13  98 101  15  1 2200  0 231 138 37  0  0]
 [  3  34  7  0  59 101  36 120 103 20  0  0  0]
 [  3  2  0 15 22 16 51 71 84 19  0  0  0]
 [ 19 40 163  0 20 41 130 77 85 34 17  0  0]
 [ 55 76  1 33 13 14  7 16 64  0  0  0  0]
 [ 12  0 12 105 34  1  9  4  0 26  0  1  0]
 [ 18  6 26 18 20 18  0  8  0 30  0  0  0]
 [  0 14  0 14 14  0  0  0 14 31  0  1  0]
 [  0  0  0  0 14  7  0  0  0 39  0  0  4]
 [ 16  0  0  0  0  6  0  0  0  0 66  0 13]
 [  0  0  0  0  0 18  0 55 33  8  0 14 26]
 [  2  0  0  0  0 24 24  0 12  0 12  0  0]
 [ 30 13  0  0  0  0  0 24 12  0  0  0  0]]
```

Figura 6: Matrizes de total de ocorrências e a de apenas reprovados

Grafo da situação de um estudante ao longo de 18 semanas. O grafo que é do tipo direcional facilita ver a trajetória que ele segue ao interagir com a plataforma LoP.

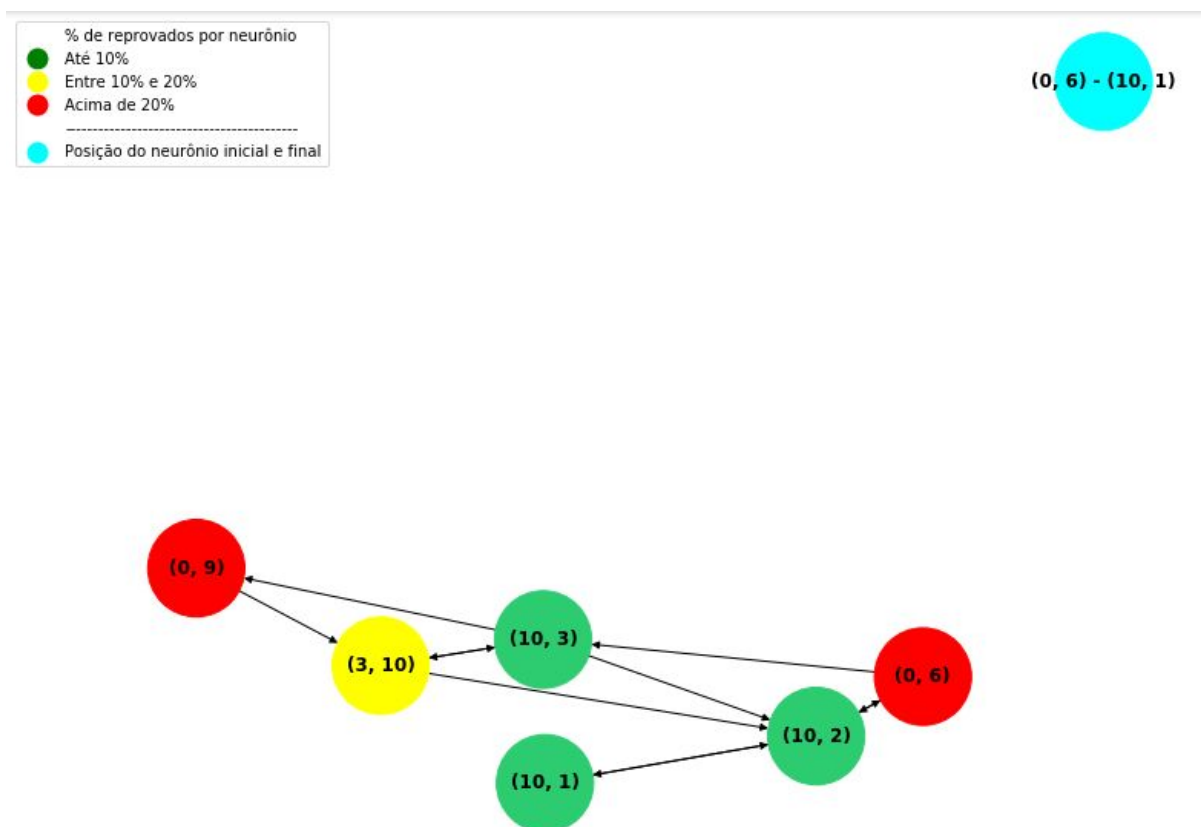


Figura 7: Grafo de um aluno

Também foi desenvolvida uma visualização para identificar os pesos de um neurônio específico, aqui vemos um exemplo de um.

```
exibeNeuronio(0,9,pesos)
```

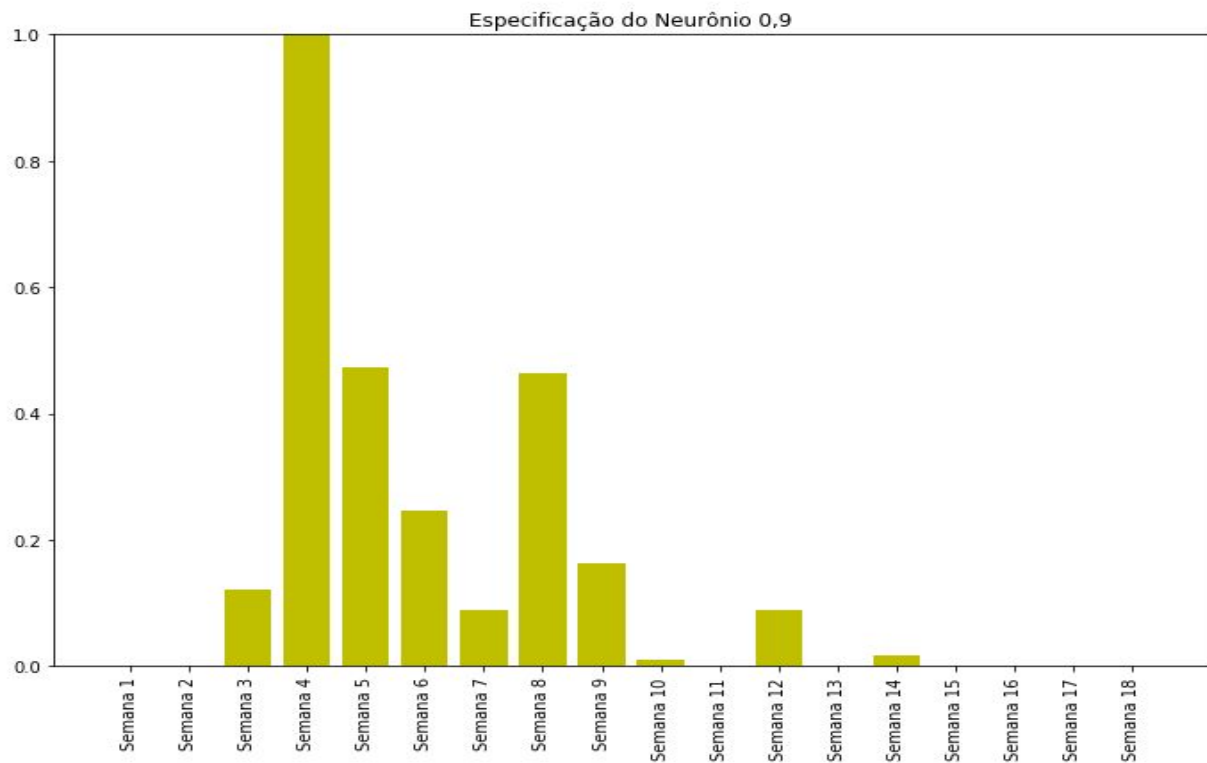


Figura 8: Especificação de um neurônio

O gráfico de superfície permite a visualização das regiões onde tem a maior concentração de reprovados.

```
chamaGraficoDeSuperficie(MContRe)
```

Gráfico de superfície - Reprovados

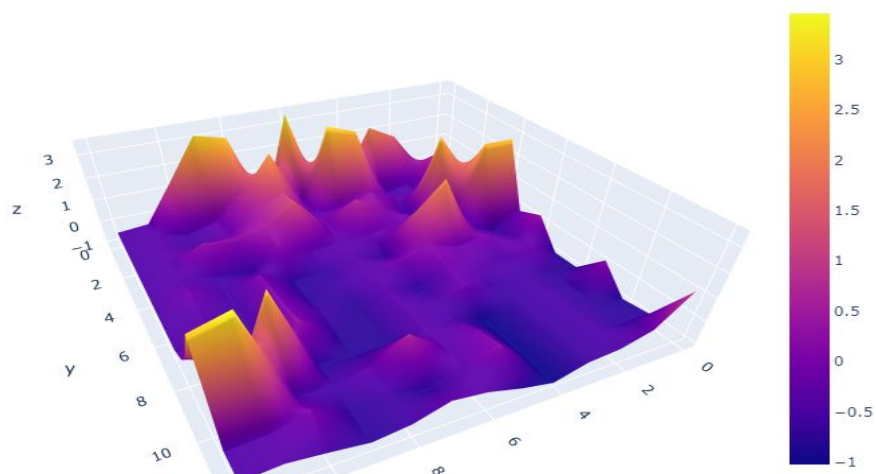


Figura 9: Gráfico de superfície com ocorrência de reprovados

E por fim, o gráfico de pizza mostrando como cada neurônio está dividido entre aprovados e reprovados.

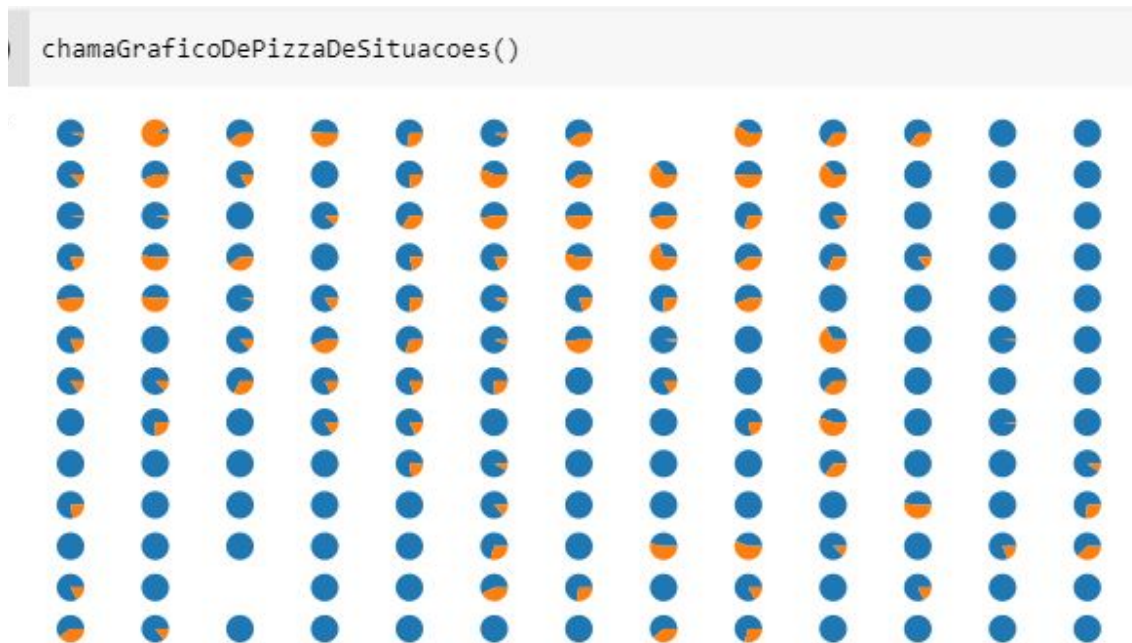


Figura 10: Gráfico de pizza com distribuição das porcentagens de reprovados e aprovados em cada neurônio

Este trabalho é uma tentativa de entender os perfis de reprovação dos alunos. Dado os resultados obtidos, pode-se dizer que foi satisfatório mas pode evoluir bem mais ainda. A rede SOM trabalhou bem em dimensionar em 2D o que era um problema em 18D. Permitindo assim, melhorar o processo de visualização dos resultados.

Link: <https://github.com/vilsonrodrigues/PegasusProject>