

Redes Auto-Organizáveis: Mapas de Kohonen

Gustavo Scaloni Vendramini
Guilherme José Henrique
Sean Carlisto de Alvarenga
Vinícius Fernandes de Jesus

10 de setembro de 2013

Resumo

Este artigo apresenta uma implementação de um modelo de rede neural auto-organizável conhecida como mapas auto-organizáveis de Kohonen ou redes SOM (do inglês *Self-Organizing Maps*).

1 Introdução

2 Redes SOM

Mapas auto-organizáveis ou Mapas de Kohonen são um tipo de rede neural. Foram desenvolvidos em 1982 por Tuevo Kohonen, professor emérito da Academia da Finlândia. SelfOrganizing mapas são apropriadamente chamado. "Auto-organização" é porque a supervisão não é necessária. SOMs aprendem por conta própria através da aprendizagem competitiva não supervisionada. "Mapas" significa que eles tentam mapear seus pesos para entrar em conformidade com os determinados dados de entrada. Os nós de uma rede SOM tentam tornar-se igual às entradas que lhes são apresentados. Neste sentido, é assim que eles aprendem. Especificamente, o relacionamento topológico das entradas de entradas topológicos são conservados quando mapeado para uma rede SOM. Isso tem um valor sensato para a representação de dados complexos.

2.1 Estrutura

A Figura ?? é uma rede SOM 4x4 (4 nós para baixo, quatro nós de diâmetro). cada mapa está ligado a cada nó de entrada. Para este pequeno nó 4x4 de rede, resulta em $4 \times 4 \times 3 = 48$ ligações. Em segundo lugar, perceber que nós do mapa não são ligados uns aos outros. Os nós são organizados desta maneira, como uma grade de duas dimensões faz com que seja fácil de visualizar os resultados. Esta representação é também útil quando o Algoritmo SOM é usado. Nesta configuração, cada nó tem um única coordenada (i, j). Isto torna mais fácil de fazer referência a um nó na rede, e calcular as distâncias entre os nós. Devido às ligações apenas para o nós de entrada, os nós mapa são indiferentes quanto ao valor que os seus vizinhos têm. Um nó de mapa só vai atualizar seus pesos "(explicado a seguir), com base no que o vetor de entrada fornece de informação.

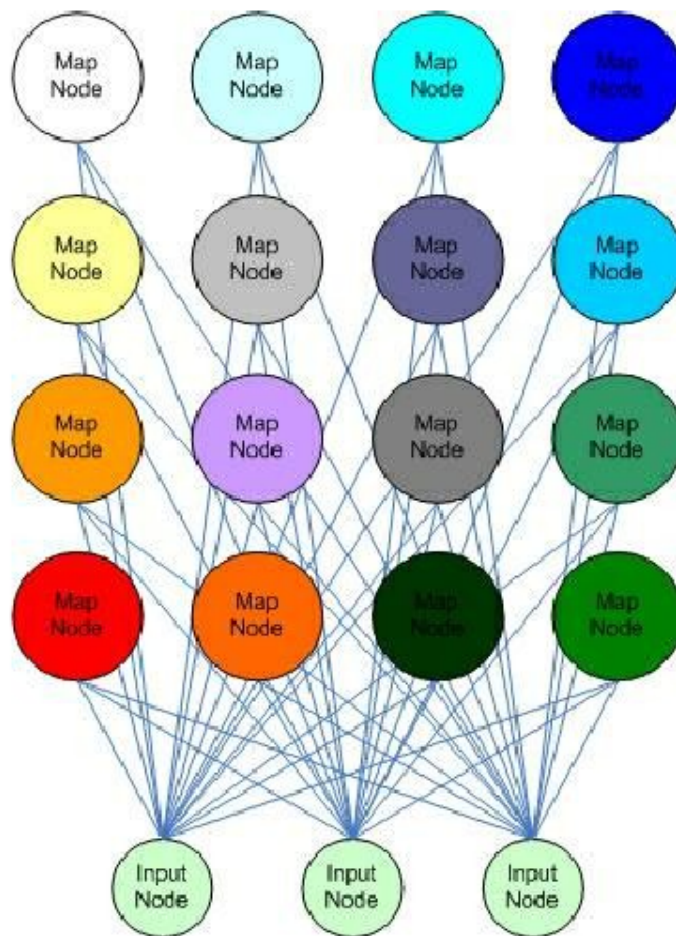


Figura 1: Nó do Mapa de Kohonen

2.2 Treinamento

A rede SOM utiliza um algoritmo de aprendizado competitivo e não-supervisionado. para o aprendizado da rede. Um padrão de entrada é apresentado a rede, um neurônio vence e inicia a atualização dos pesos do neurônio vencedor e de seus vizinhos (até um raio de vizinhança). Isto repete para cada nova entrada e a taxa de aprendizado e o raio de vizinhança são decrementados durante o processo.[Braga]

Atualização dos pesos do neurônio vencedor e de seus vizinhos:

$$w_{ji}(t+1) = \begin{cases} w_{ji}(t) + \eta(t)(x_i - w_{ji}(t)) & \text{se } j \in \Lambda(t) \\ w_{ji}(t) & \text{caso contrário} \end{cases}$$

Onde w_{ji} é o peso entre os neurônios i e j , $\eta(t)$ é a taxa de aprendizado e Λ é a vizinhança.

```

1: Inicializar pesos e parâmetros;
2: repita
3:   para todo padrão de treinamento faça
4:     Definir neurônio vencedor;
5:     Atualizar os pesos deste neurônio e de seus vizinhos;
6:     se o número do ciclo for múltiplo de N então
7:       Então reduzir a taxa de aprendizado;
8:     fim-se
9:   fim-para
10: até que mapa de características não mudar
```

2.3 Classificação de Cor

Este é considerado o "Hello World" de SOMs. A maioria dos tutoriais publicados no SOMs usam a classificação de cor SOM como seu principal exemplo. Isto é para uma muito boa razão. Ao passar por este exemplo, o conceito de SOMs pode ser solidamente apreendido. A razão pela qual a classificação de cor é bastante fácil de entender é por causa da quantidade relativamente pequena de dados utilizados, bem como o aspecto visual dos dados.

A classificação de cores do SOM usam apenas três pesos por mapa e nós de entrada. Estes pesos representam a tripla (r, g, b) para a cor. Por exemplo, as cores podem ser apresentado à rede, (1,0,0) para o vermelho (0,1,0) para o verde, etc A meta para a rede aqui, é para aprender a representar todas essas cores de entrada em sua grade de duas dimensões, mantendo as propriedades intrínsecas de tal como mantendo as relações topológicas entre os vetores de entrada. Com isso em mente, se azul escuro e azul claro são apresentados ao SOM, eles devem acabar ao lado do outro da grelha de rede.

Para ilustrar o processo passo a passo através do algoritmo para a aplicação de classificação da cor. O primeiro passo representa a inicialização da rede. A Figura 2 mostra uma rede recém inicializado. Cada quadrado representa um nó da rede.

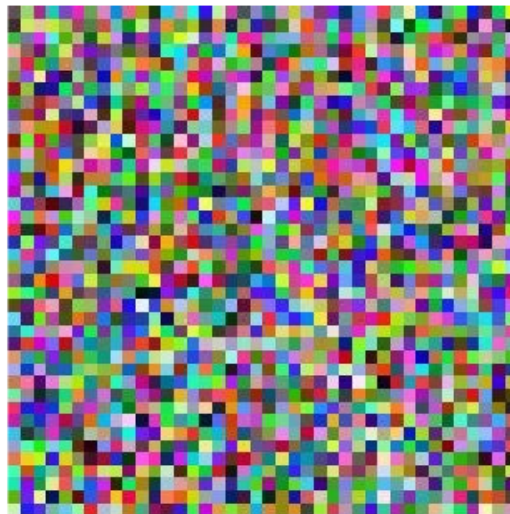


Figura 2: Disposição inicial dos neurônios

O segundo passo é escolher um vetor de forma aleatória a partir dos vetores de entrada. O passo 2 escolhe um vetor de forma aleatória a partir dos vetores de entrada. Oito vetores de entrada são utilizados neste exemplo, que vão do vermelho para o amarelo para verde escuro. Em seguida, o Passo 3 passa por cada nó e encontra BMU, conforme descrito anteriormente. A Figura 3 mostra o BMU sendo selecionado na rede 4x4. A etapa 4 do algoritmo calcula o raio vizinhança. Isto também é mostrado na Figura 3. Todos os nós em vermelho estão dentro do raio. No passo 5, em seguida, aplica-se o aprendizado funções

para todos estes nodos. Baseia-se na sua distância a partir do neurônio vencedor. O neurônio vencedor (vermelho escuro) aprende mais, enquanto nós, na periferia do raio (luz rosa) aprendem o mínimo. Nós fora do raio (branco) não aprendem nada.

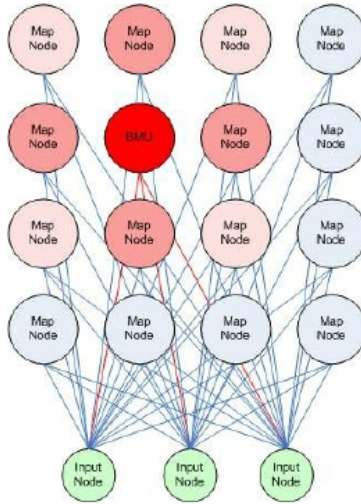


Figura 3: Nó vencedor

Em seguida, volte para a Etapa 2 e repita. A Figura 4 mostra um SOM treinado, representando todas as cores de oito entradas de cores. Observe como a luz verde fica ao lado do verde escuro, e vermelho está ao lado de laranja.

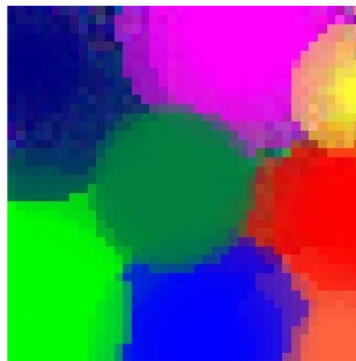


Figura 4: Estrutura do nó final

3 Métodos

4 Resultados

A rede neural auto-organizável implementada nesse trabalho, utilizou o algoritmo Mapa de Kohonen, desenvolvido por Teuvo Kohonen em 1982 [1]. A implementação deste trabalho pode ser vista em <https://github.com/QSF/ia-kohonen>.

Para os resultados práticos, a rede foi treinada com um conjunto de 1728 exemplos de dados reais, fornecidos pelo site *UCI Machine Learning Repository*¹, variando valores da taxa de aprendizado, pesos iniciais, raio da vizinhança e quantidade de neurônios na rede. O conjunto de exemplos escolhido foi o *Car Evaluation*², onde dados os atributos preço de compra do veículo, custo da manutenção, quantidade de portas do veículo, número de passageiros, tamanho do bagageiro e segurança estimada do veículo, avalia se o veículo é aceitável, inaceitável, bom ou muito bom.

Uma vez que o algoritmo Mapa de Kohonen organiza ou agrupa dimensionalmente os dados, o conjunto de exemplos foi adaptado, apresentando somente três dimensões (preço de compra do veículo, custo da manutenção e segurança estimada do veículo). O primeiro e segundo atributo aceitos os valores low, med, high e vhigh, enquanto que o útil aceita os valores low, med, high.

Com o conjunto de dados coletados, algumas entradas foram testadas para verificar a modificação final da rede. Esses testes serão exibidos a seguir, sendo dividido pela quantidade de neurônios na camada de saída (16, 20, 25 e 36 neurônios).

4.1 16 neurônios

Nos conjuntos de teste apresentados por essa seção, a rede possuía 16 neurônios na camada de saída, sendo que sua topologia é da forma de uma grade 4x4.

Para realizar os testes na rede mencionada, foi variado o valor da taxa de aprendizagem e dos pesos iniciais, com raio fixo em 0.

Quatro testes foram executados, onde a configuração da rede para cada caso é ilustrada em 1.

Tabela 1: Configurações da Rede com 16 Neurônios na Saída

Teste	Taxa de Aprendizagem	Pesos Iniciais
1	0.1	todos 1
2	0.1	Random
3	0.3	todos 1
4	0.3	Random

As figuras 5, 6, 7 e 8 representam a rede após a execução dos testes 1, 2, 3 e 4; respectivamente.

4.2 20 neurônios

Nesta seção, a rede possuía 20 neurônios na camada de saída, sendo uma grade 4x5.

¹<http://archive.ics.uci.edu/ml/>

²<http://archive.ics.uci.edu/ml/datasets/Car+Evaluation>

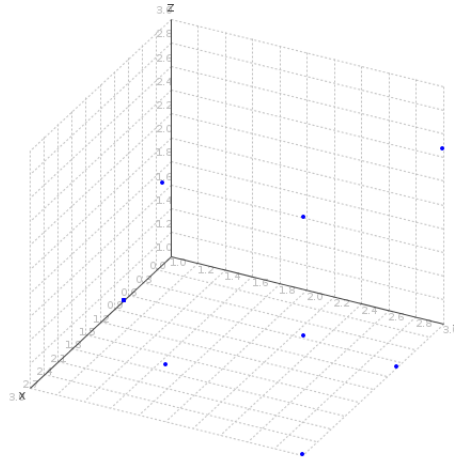


Figura 5: 16 Neurônios - Teste 1

Para realizar os testes na rede mencionada, novamente foi variado o valor da taxa de aprendizagem e dos pesos iniciais, mas com raio fixo em 1.

Também foram executados quatro testes que tem sua configuração ilustrada na tabela 2.

Tabela 2: Configurações da Rede com 20 Neurônios na Saída

Teste	Taxa de Aprendizagem	Pesos Iniciais
1	0.5	todos 1
2	0.5	Random
3	0.75	todos 1
4	0.75	Random

Os testes 1, 2, 3 e 4 são representados pelas imagens 9, 10, 11 e 12; respectivamente.

4.3 25 neurônios

Os gráficos a seguir mostram a plotação dos pontos após a execução do algoritmo para alguns valores de parâmetros e para uma rede com 25 neurônios. Apesar de não se poder afirmar se o algoritmo convergiu, vemos que aumentando o raio de vizinhança, o algoritmo consegue “especializar” melhor o conjunto. O efeito da taxa de aprendizado está sobre a velocidade na qual o algoritmo irá convergir. Ela geralmente é dada empiricamente, assim como os pesos iniciais, que segundo

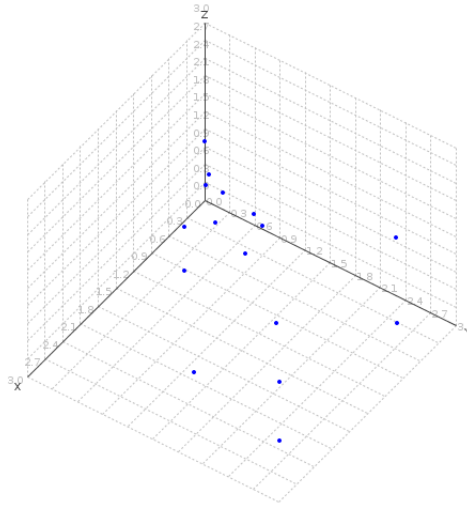


Figura 6: 16 Neurônios - Teste 2

[1], se forem inicializados com valores iguais, faz o algoritmo convergir mais rapidamente.

4.4 36 neurônios

Os gráficos a seguir mostram a plotação dos pontos após a execução do algoritmo para alguns valores de parâmetros e para uma rede com 36 neurônios.

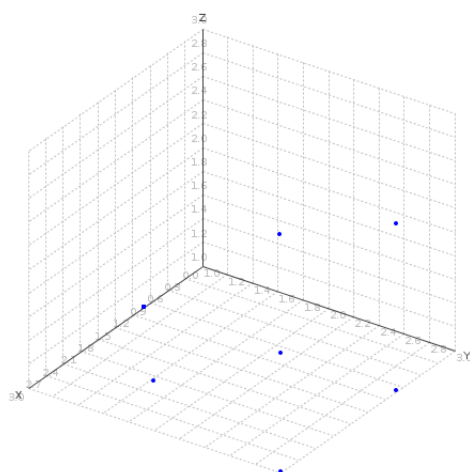


Figura 7: 16 Neurônios - Teste 3

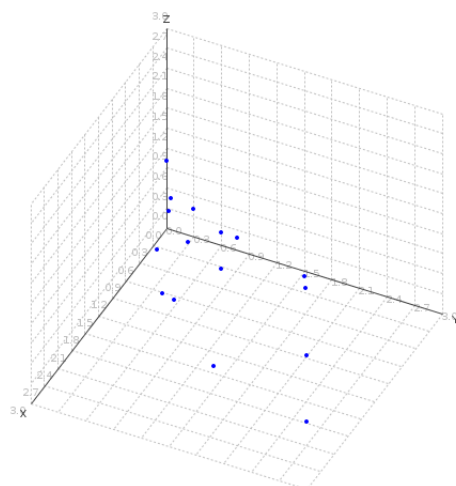


Figura 8: 16 Neurônios - Teste 4

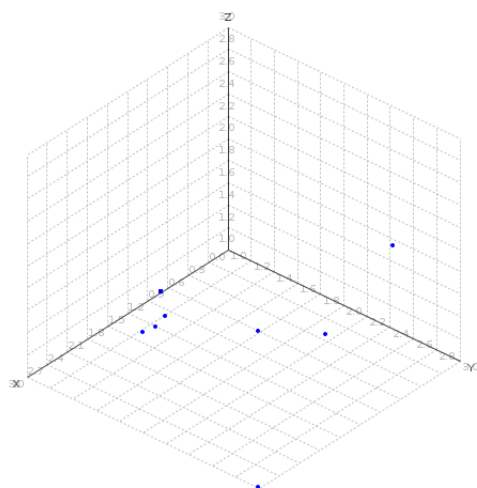


Figura 9: 20 Neurônios - Teste 1

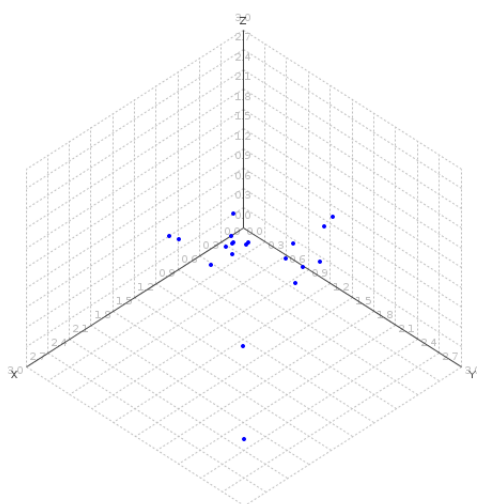


Figura 10: 20 Neurônios - Teste 2

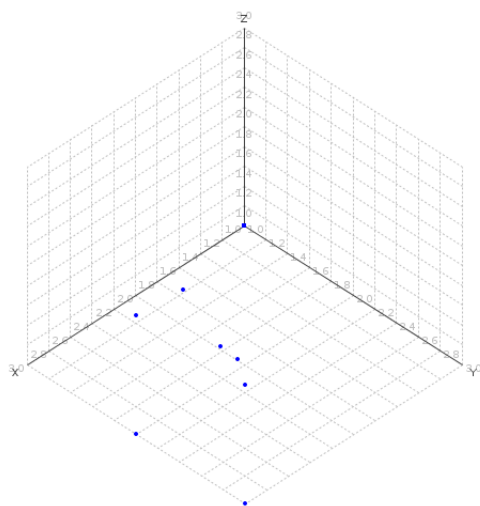


Figura 11: 20 Neurônios - Teste 3

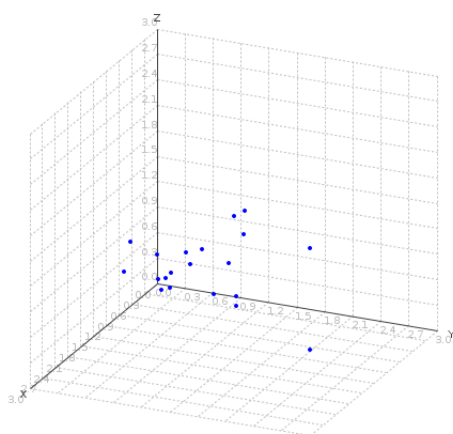


Figura 12: 20 Neurônios - Teste 4

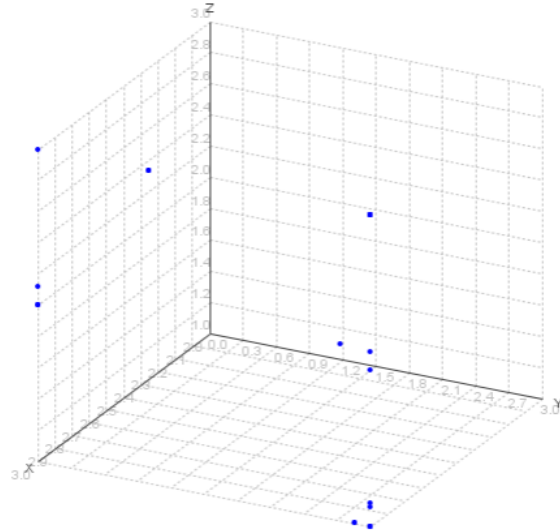


Figura 13: Taxa de aprendizado: 0.8662815917277779; Raio: 1; Pesos iniciais: A

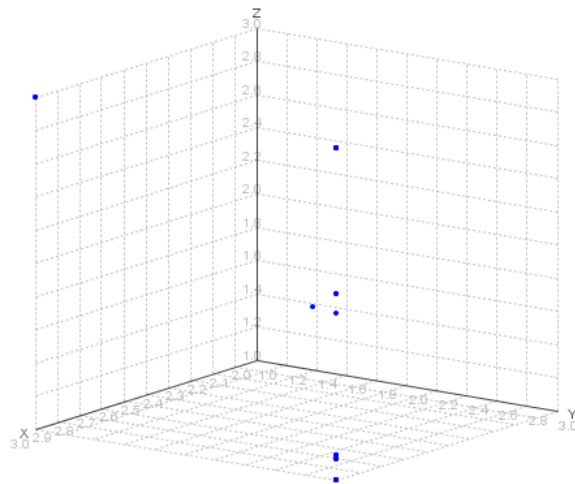


Figura 14: Taxa de aprendizado: 0.8662815917277779; Raio: 2; Pesos iniciais: A

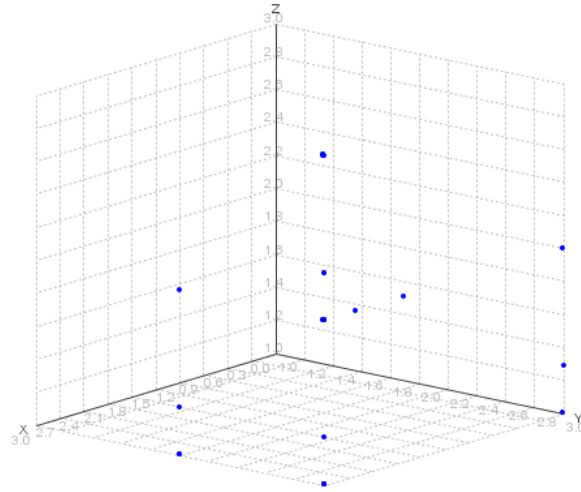


Figura 15: Taxa de aprendizado: 0.5980629675758204; Raio: 1; Pesos iniciais: B

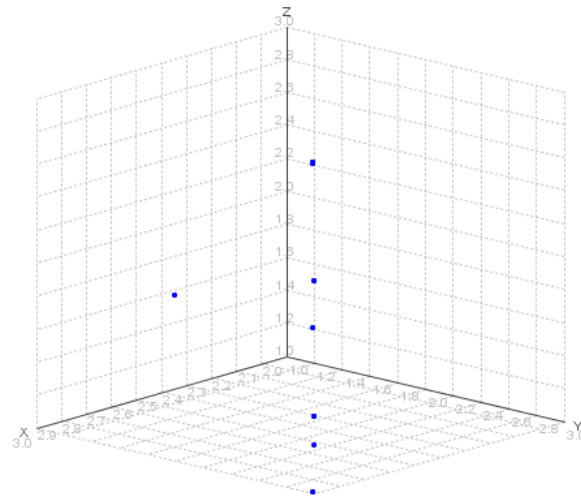


Figura 16: Taxa de aprendizado: 0.5980629675758204; Raio: 2; Pesos iniciais: B

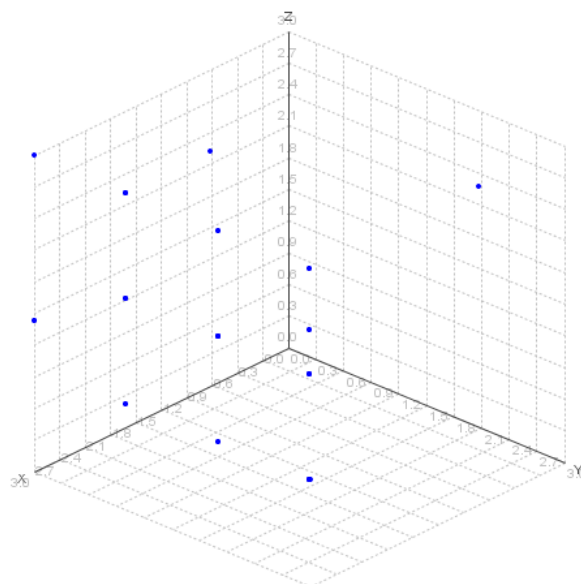


Figura 17: Taxa de aprendizado: 0.2625656059885696; Raio: 1; Pesos iniciais: D

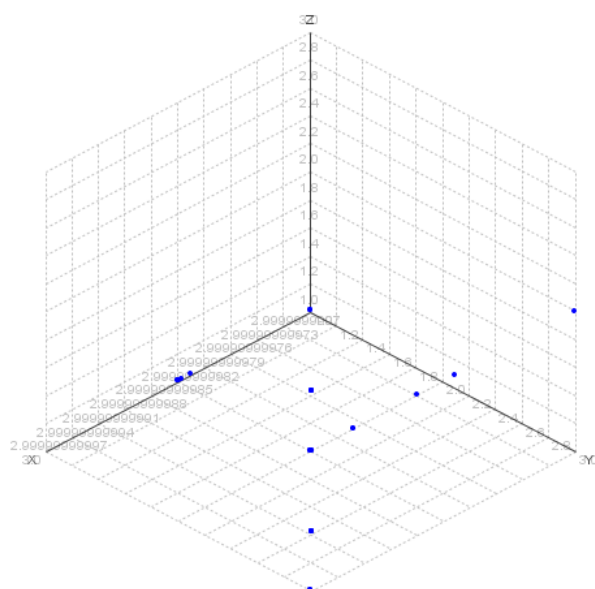


Figura 18: Taxa de aprendizado: 0.2625656059885696; Raio: 2; Pesos iniciais: D

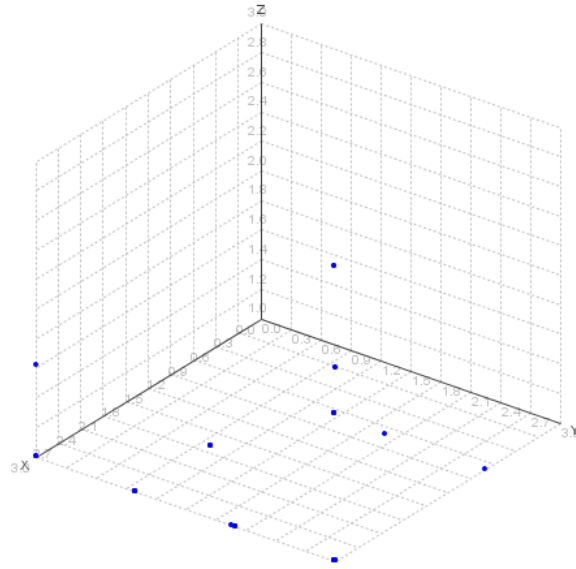


Figura 19: Taxa de aprendizado: 0.5517820314689292; Raio: 1; Pesos iniciais: E

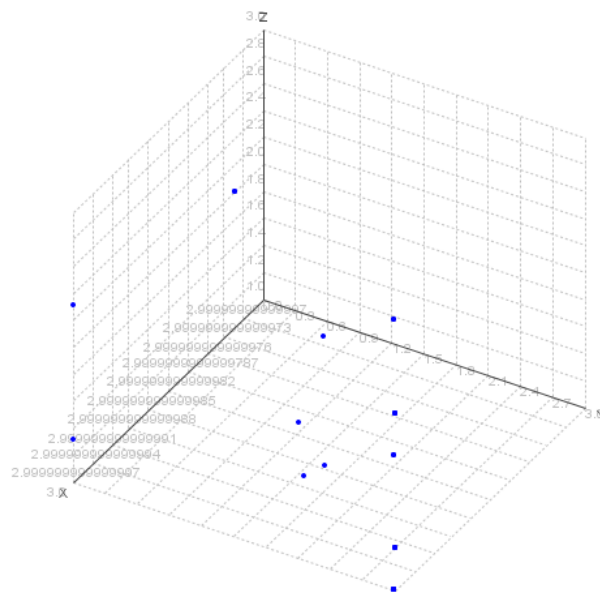


Figura 20: Taxa de aprendizado: 0.5517820314689292; Raio: 2; Pesos iniciais: E

5 Conclusão

Referências

- [1] Teuvo Kohonen. *Self-Organizing Maps*. Springer, 2000.