

Sistema de Informação Inteligente para Análise de Crédito

Alan Tandler Leibel Bacellar¹ e Pedro Maciel Xavier¹

20 de setembro de 2017

¹ Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brasil

Abstract. I'd like to talk, in about 150 words, a little about the FAT-WiSARD applied to Credit Analysis

1 O Modelo *WiSARD*

Quando falamos sobre Redes Neurais sem Pesos nos referimos de alguma forma ao modelo *WiSARD* (*Wilkie, Stonham and Aleksander's Recognition Device*), que consiste em armazenar o conteúdo de um dado treinamento em estruturas análogas a memórias *RAM* (*Random Access Memory*), organizadas em discriminadores. Cada discriminador representa uma classe de padrões a ser aprendido e tem suas *RAMs* treinadas com padrões que pertencem a essa mesma classe.

Uma Rede deste tipo contém, a princípio, três parâmetros: n , o número de *RAMs*; d , que é o número de *bits* que compõem o endereço de uma it *RAM*; e b , conhecido como limiar de *bleaching*, do qual falaremos adiante. Podem ser usadas diversas arquiteturas diferentes, ou seja, podemos escolher n e d de forma a obter melhores resultados. É preferível que n e d sejam escolhidos de modo que o produto $n.d$ seja igual ao número de elementos do vetor de entrada. Caso isso não seja possível, podemos simplesmente utilizar um mapeamento que não inclua alguma entrada.

É feito, previamente, um mapeamento, em geral aleatório, que define quais elementos de um dado vetor de entrada vão compor o endereço a ser acessado por uma *RAM*. As entradas da rede, no entanto, devem ser previamente convertidos para a forma binária, ou em um caso geral, simplesmente discretizados para um intervalo finito.

$$entrada = (e_1, e_2, e_3, \dots, e_{n.d})^T \quad e_i \in \{0, 1\} \quad \forall i$$

$$mapeamento = (7, 4, \dots, n.d, \dots, 1, 9)^T$$

$$entrada' = (e_7, e_4, \dots, e_{n.d}, \dots, e_1, e_9)^T$$

Em seguida as entradas que passaram pelo mapeamento são agrupadas conforme o parâmetro d para assim constituírem o "vetor endereço" que será acessado em cada *RAM*.

$$entrada'' = ((e_7, e_4, \dots)^T, \dots, (\dots, e_{n.d}, \dots)^T, \dots, (\dots, e_1, e_9)^T)$$

Seja a_i o endereço a ser acessado na i -ésima *RAM*, podemos obtê-lo como um índice tomando o produto interno entre o vetor dos componentes do endereço com um vetor das potências de 2 (ou da base que estiver em uso) de mesmo comprimento. Podemos também usar simplesmente o "vetor endereço" como endereço para acessar as posições da *RAM*.

$$a_i = \langle (e_\alpha, e_\beta, e_\gamma \dots)^T \mid (2^0, 2^1, 2^2, \dots)^T \rangle$$

$$entrada''' = (a_1, a_2, a_3, \dots, a_n)$$

Dessa forma se atribui valor 1 a cada endereço que fora ativado durante o processo de treinamento, enquanto que aqueles que não o foram permanecem relacionados com o valor inicial 0. Também é possível incrementar o valor armazenado toda vez que uma determinada posição for acessada.

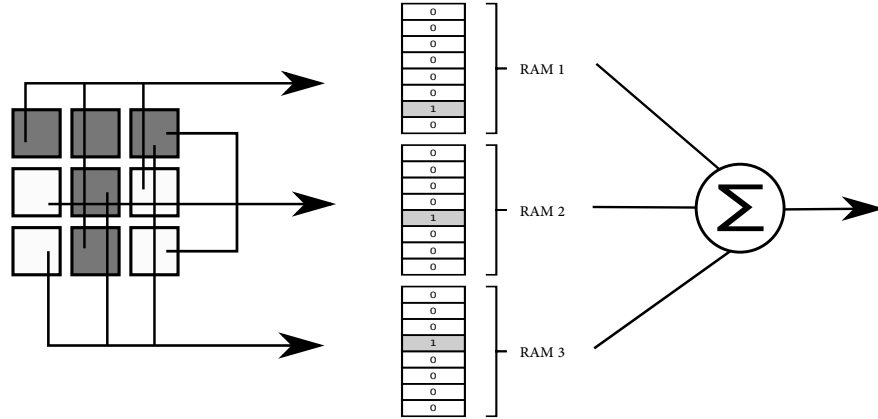


Figure 1: WiSARD

Quando passamos a tentar reconhecer padrões já treinados, basta repetir o mesmo processo de endereçamento, mas agora efetuando uma leitura do valor armazenado no endereço de memória acessado. Assim, quando avaliamos o somatório das leituras realizadas em todas as *RAMs* de um dado discriminador, obtemos a pontuação daquela classe quando exposta a um padrão desconhecido. Em seguida basta decidir por aquela classe cujo discriminador obteve a maior pontuação dentre os demais.

Concluído o processo, podemos ainda nos deparar com casos em que duas ou mais classes apresentem pontuações muitíssimo próximas, o que nos levaria a recorrer a algum método de desempate. Um método que se mostra muito efetivo é o *bleaching* (Citar o PESC) que consiste em considerar apenas as saídas das memórias que obtiverem como resultado da leitura um valor maior que b .

2 Neurônio com Peso

O neurônio com peso possui variáveis, chamados de pesos, que influenciam na ativação do neurônio. Os pesos vão sofrendo alterações durante a etapa de treinamento, o que muda assim, a resposta dada pelo neurônio.

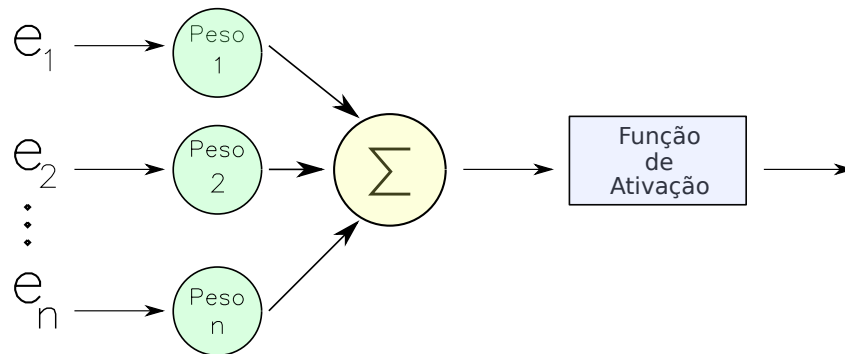


Figure 2: Arquitetura do discriminador

3 *FAT-WiSARD*

Após o entendimento de uma rede *WiSARD*, pode-se notar que a mesma trata todas as *RAMS* de um discriminador com a mesma importância, ou seja, caso uma delas seja ativada, ela contribuirá com a mesma pontuação no discriminador comparado a outras *RAMS* que também ativaram. Isso nem sempre é desejado, pois um conjunto de *features* pertencente a uma *RAM* pode ter importância maior no reconhecimento do padrão que outros conjuntos. A *FAT-WiSARD* propõe uma solução para esse problema, adicionando um neurônio com peso, na saída de cada *RAM*, no qual esse determinará o valor que aquela *RAM* tem para o reconhecimento do padrão.

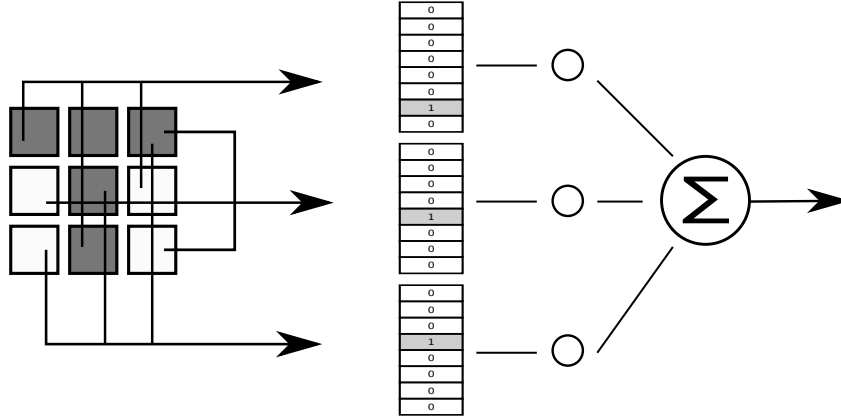


Figure 3: Arquitetura do discriminador

Outro problema que a *WiSARD* apresenta, análogo ao citado acima, é que as *features* que endereçam uma *RAM* sempre têm a mesma importância na ativação da mesma. Assim, para contornar esse problema, é adicionado uma camada de neurônios com pesos, nas entradas de cada *RAM*.

Porém, com a adição dessa camada pesos prévias as *RAMS*, um novo problema é gerado. Agora, as *RAMS* não estão mais recebendo um valor discreto, necessário para seu funcionamento. Com isso em vista, é proposta uma substituição das *RAMS* por algo de mesma funcionalidade e que aceite uma entrada contínua. Para isso é utilizado um espaço euclidiano, que ao invés de incrementar o número contido no endereço fornecido pela entrada,

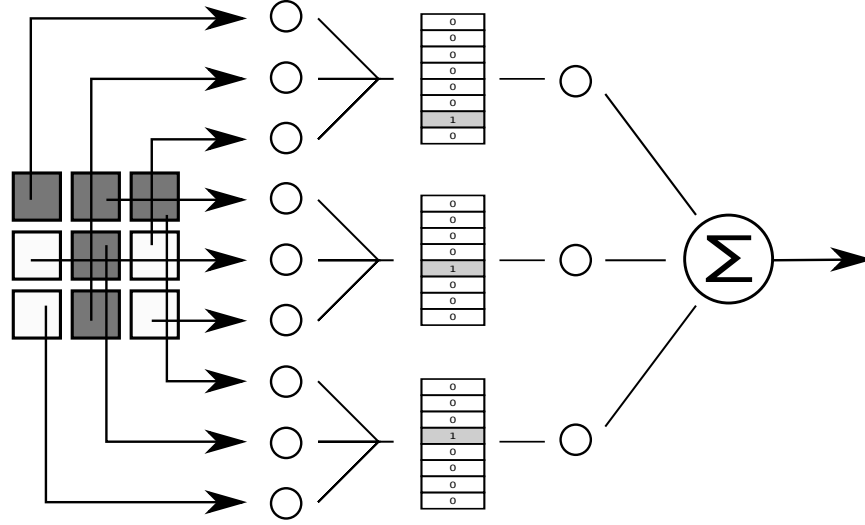


Figure 4: Arquitetura do discriminador

será guardado o vetor de entrada como um ponto nesse espaço. Sua pontuação será dada pelo soma do número de pontos contidos no espaço que possui uma distância euclidiana igual a zero em relação ao ponto de entrada. Assim, esse espaço possuirá uma resposta igual a resposta de uma *RAM*.

Para um melhor funcionamento do espaço, leva-se em conta os ruídos contidos nas entradas, que podem levar ao não reconhecimento da entrada caso presentes. Assim, defini-se uma peso r , que dará uma pontuação maior se houver pontos próximos aos ponto a ser reconhecido, e que esteja dentro de um raio determinado pela peso.

Trocando assim as *RAMS* pelos espaços euclidianos, temos a *FAT-WiSARD*. O treinamento na *FAT-WiSARD* é uma junção do treinamento da *WiSARD* com os de neurônios com peso. Primeiro realiza-se o treinamento igual ao da *WiSARD*, em seguida, realiza-se uma atualização dos pesos (*e.g. Gradient Descent*), e para finalizar, apaga-se todo o conteúdo salvo nos espaços e realiza novamente o treinamento estilo *WiSARD*, finalizando assim um processo iterativo do treinamento.

4 Análise de Crédito

Segundo a Figura 3, ...

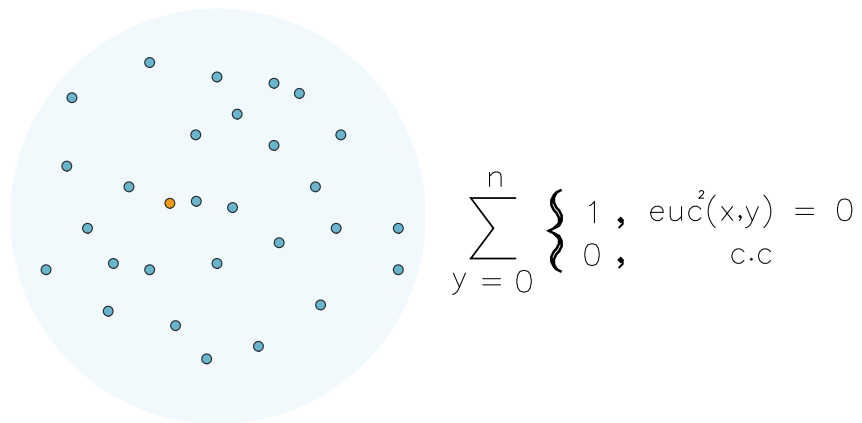
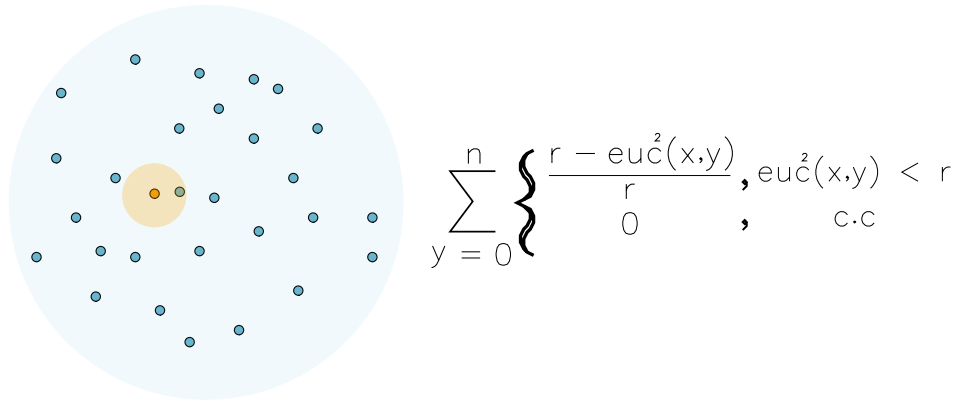


Figure 5: Espaço Euclidiano

Aqui nao tem nada mesmo.



$$\sum_{y=0}^n \left\{ \begin{array}{l} \frac{r - \text{eucl}^2(x,y)}{r}, \text{eucl}^2(x,y) < r \\ 0, \text{c.c} \end{array} \right.$$

Figure 6: Espaço Euclidiano

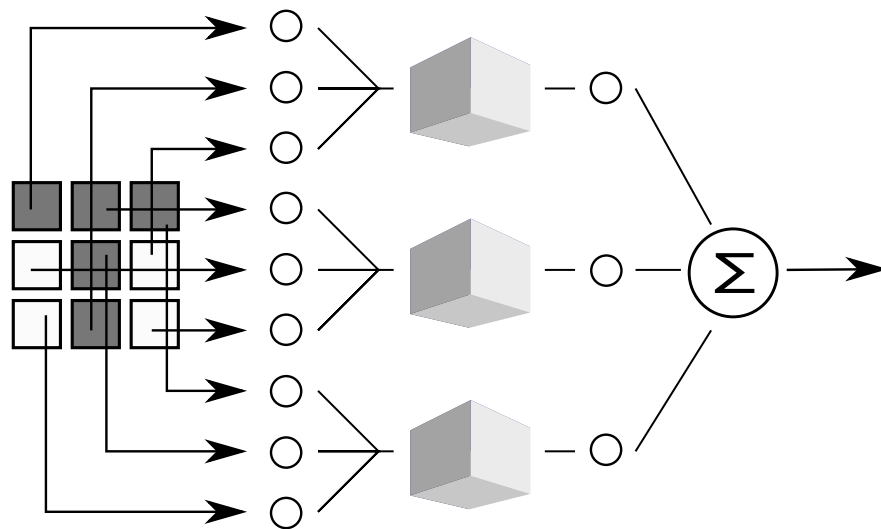


Figure 7: Espaço Euclidiano