

Autenticação Bancária via Sistemas Fuzzy x Redes Neurais

John Theo S. de Souza¹

¹Engenharia da Computação – Universidade Tecnológica Federal do Paraná
Av. Sete de Setembro, 3165 – Curitiba – PR – 80230-901

john.theo.souza@gmail.com.br

Resumo. Este documento apresenta uma comparação entre um classificador fuzzy e uma Rede Neural MLP (Multi Layer Perceptron) para o problema de classificação de cédulas monetárias

Palavras-chave: fuzzy, rede neural, classificador.

1. Introdução

O problema trata-se da classificação de cédulas monetárias. Dada uma imagem de uma cédula é obtida realiza-se a transformada Wavelet desta para obter os seguintes parâmetros:

- variance
- skewness
- curtosis
- entropy

Guarda-se esses parâmetros juntamente com um valor de saída, definindo se a célula é autêntica(0) ou falsificada(1). Sendo assim, cada tupla de dados fica da seguinte forma: (variance, skewness, curtosis, entropy, saída)

Um conjunto de dados nesse formato foi disponibilizado em um arquivo para que pudesse ser utilizado nas abordagens Fuzzy e Redes Neurais. Este conjunto de dados foi particionado em duas partes: uma para treinamento(80% dos dados) e outra para teste(10% dos dados)

2. Classificador Fuzzy

Classificação Fuzzy é o processo de agrupar elementos em conjuntos fuzzy onde a função de pertinência é definido pela validação de uma função proposicional fuzzy [Zimmermann 2012]

A modelagem para o classificador ao qual esse trabalho destina-se foi dividida em 3 partes: Base de dados, Treinamento / Base de Regras e Teste.

2.1. Base de Dados

A partir dos dados fornecidos, separou-se os cada variável em um conjunto de dados para que assim fosse possível separar os universos de discurso de cada parâmetro. Para fazer esse definição utilizou-se o máximo e o mínimo de cada conjunto e um offset de duas unidades para não perder informação nos extremos.

Tendo o universo de discurso definido, é possível gerar as variáveis linguísticas fuzzy, que nada mais são do que funções de pertinência(membership function). Este é um

passo importante pois é onde discretiza-se o universo de discurso em conjuntos fuzzy. Primeiro pois é necessário ter um conhecimento prévio sobre as variáveis para poder separar as classes/níveis de forma coerente e um maior numero de variáveis implica em um numero maior de regras. Para particionar universo, optou-se pela abordagem uniforme onde o universo da variável será dividido em 4 partes, ou seja, cada variável terá um função baixa(lo), média baixa(ml), média alta(mh) e alta(hi). Cada uma será uma função triangular e os pontos de inicio, meio e fim serão obtido como frações do limite máximo. Feito isso, os conjuntos ficaram divididos conforme a Figura1

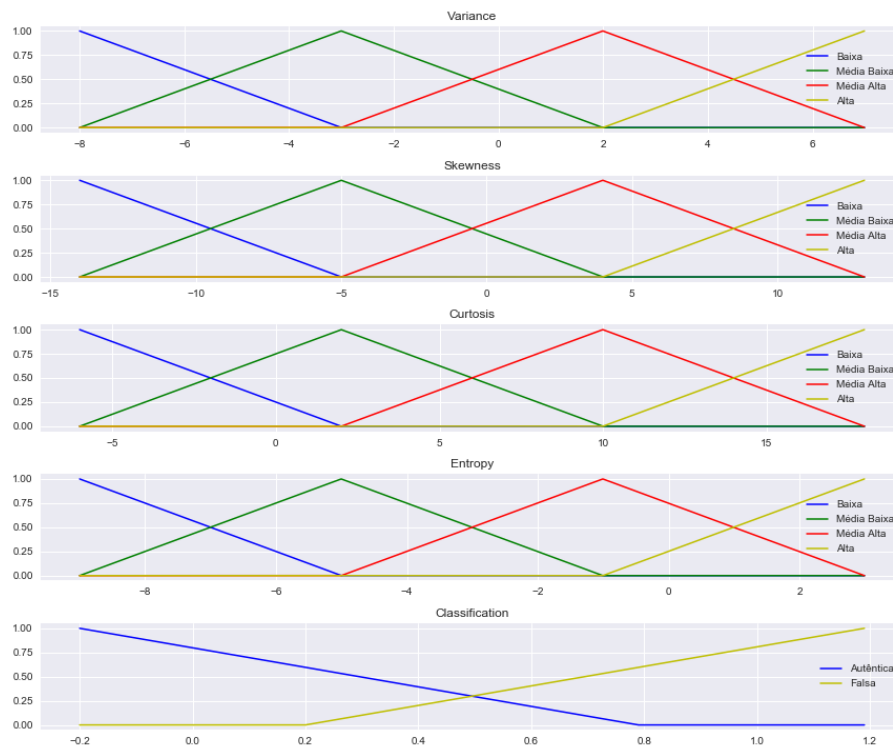


Figura 1. Conjuntos Fuzzy

Com base do histograma de cada variável, montou-se uma segunda configuração das variáveis buscando concentrar os conjuntos onde havia maior incidência dos dados, porém mantendo as 4 variáveis linguísticas(baixa, média baixa, média alta, alta).

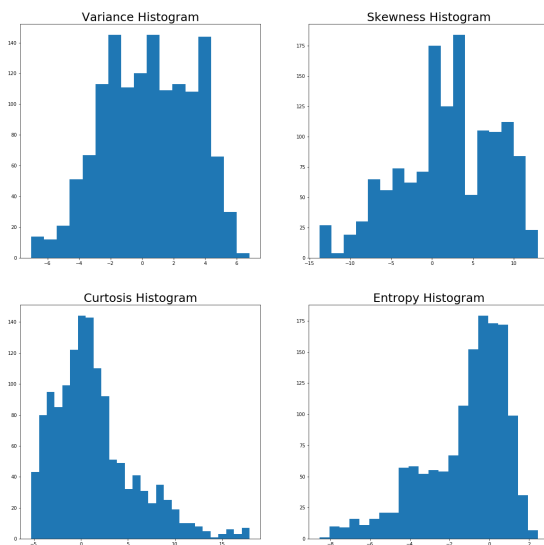


Figura 2. Histograma das variáveis

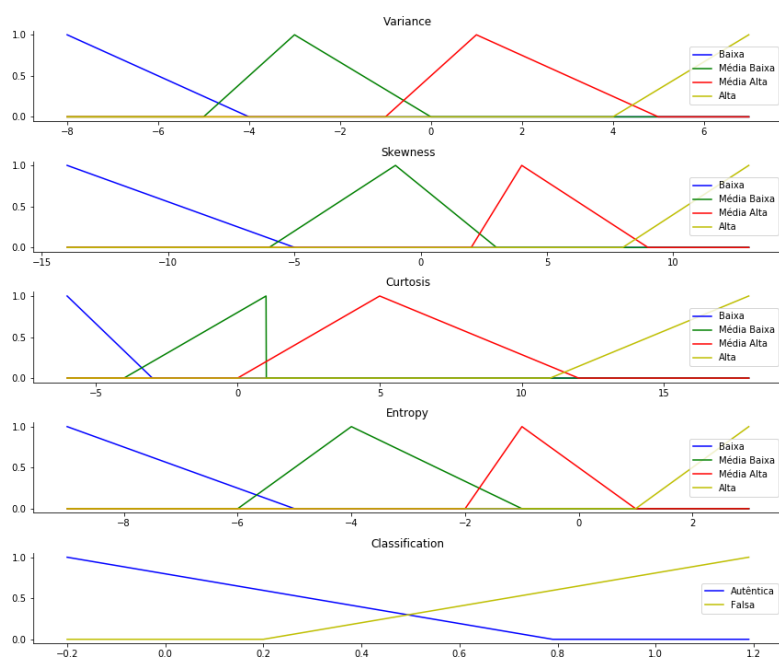


Figura 3. Conjuntos Fuzzy com base no histograma

2.2. Treinamento/Base de Regras

Para montar a base de regras utilizou-se o método de Wang-Mendel[Wang and Mendel 1992] onde ela é gerada a partir do processo de treinamento. Com os valores de treinamento, aplicou-se um a um a cada conjunto fuzzy para obter o nível de ativação referente àquele conjunto. Com o valor de ativação para cada

variável, aplica-se a norma-T e insere em uma base de regras da seguinte forma: variáveis_ativadas:(valor_ativacao,classe_treino), por exemplo: ('Variance Alta', 'Skewness Alta', 'Custosis Baixa', 'Entropy Media Alta'): (0.507, 0.0). Ao inserir nessa base de regras, verifica-se se já existe alguma regra com os mesmos precedentes, se sim, compara-se o valor_ativacao e persiste o de maior valor. Ao final desse processo, tem-se uma base de regras gerada.

Para a configuração uniforme dos conjuntos, obteve-se uma média de 46 regras e para a configuração baseada no histograma, obteve-se uma média de 58 regras.

Vale ressaltar que como o conjunto de dados é embaralhado antes de fazer a separação de conjuntos de teste e treinamento, existe a possibilidade de a base de regras não ser sempre igual, devido à dependência que o método de Wang-Mendel tem com a amostragem dos dados.

2.3. Teste

Tendo a Base de Regras estabelecida, é possível utilizar os dados de teste e realizar a inferência fuzzy. Para este trabalho, no que diz respeito a defuzzificação utilizou-se o cálculo por centro de massa pois as variáveis de saída era apenas duas e assim não haveria problema de inferir um conjunto errado caso precisasse decidir entre dois conjuntos com peso alto.

Foram feitas 100 iterações onde treinava-se o modelo e depois o testava para cada uma das configurações. Ao final obteve-se um acerto de aproximadamente 58% para a configuração uniforme e 60% para a configuração baseada no histograma.

2.4. Considerações

A falta de um especialista pode estar relacionada com a baixa assertividade do classificador fuzzy ao passo que particionando o conjunto uniformemente, pode-se ter particionado um intervalo do universo de discurso menos do que deveria. Com o particionamento baseado no histograma, percebeu-se uma melhora pequena, mas que indica que um melhor particionamento levando em conta a configuração dos dados pode sim propiciar um classificador com melhor assertividade. Outra abordagem possível seria particionar o universo de discursos de cada variável em mais partes, mas isso implica em um aumento exponencial do número de regras possíveis, porém com o modelo do Wang-Mendel, não seria necessário mapear todas as regras.

3. Rede Neural

A implementação da Rede Neural foi feita utilizando a biblioteca MLPClassifier do pacote sklearn[Buitinck et al. 2013] que usa forward propagation para calcular o estado da rede e a partir daí a função de custo, e usa back propagation como um passo para calcular as derivadas parciais da função de custo.

Ao criar uma RN utilizando o MLPClassifier, temos os seguintes parâmetros:

- hidden layer size - todos os layers tem a mesma quantidade de neuronios e sao todos interligados a tupla diz quantos layers e quantos neuronios cada
- activation - função de ativação
- max iter - Número maximo de iterações

- solver - Função utilizada para atribuir os pesos. sgd = stochastic gradient descent
- learning rate - valor constante atribuído para atualização dos pesos. É o tamanho do passo do gradient descent
- tol - limiar tolerancia. Quando duas gerações não melhoram por pelo menos o valor de tol, a rede para de iterar inferindo que chegou a um máximo local

Com base nesses parâmetros, foi realizado alguns testes para analisar o comportamento da rede em diferentes situações.

3.1. Topologia da Rede

O primeiro passo foi realizar configurações diversas no que diz respeito ao arranjo dos neurônios.

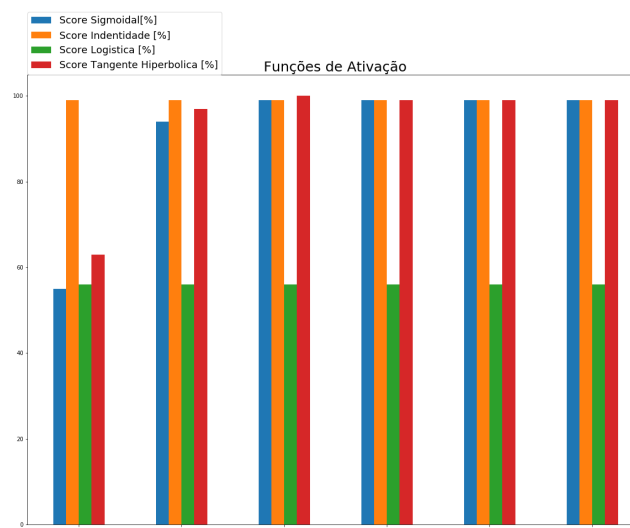
Config (layer1,layer2)	Acerto Treino (%)	Acerto Teste (%)
(1, 0)	92	93
(1, 1)	56	56
(1, 2)	61	55
(1, 3)	99	99
(1, 4)	64	58
(1, 5)	99	99
(2, 0)	99	99
(2, 1)	56	56
(2, 2)	91	94
(2, 3)	99	99
(2, 4)	99	100
(2, 5)	99	99
(3, 0)	99	99
(3, 1)	98	99
(3, 2)	99	99
(3, 3)	99	99
(3, 4)	99	99
(3, 5)	99	99
(4, 0)	99	99
(4, 1)	100	100
(4, 2)	99	99
(4, 3)	99	99
(4, 4)	99	99
(4, 5)	99	99
(5, 0)	100	100
(5, 1)	99	99
(5, 2)	99	99
(5, 3)	98	98
(5, 4)	99	99
(5, 5)	99	98
(6, 0)	100	100
(6, 1)	83	86
(6, 2)	100	100
(6, 3)	100	100
(6, 4)	100	100
(6, 5)	100	100

Figura 4. Acurácia por Topologia

Olhando para os scores, pode-se concluir que a base de dados é uma base simples(linearmente independente) pois com apenas 1 perceptron no primeiro layer já é possível obter uma acurácia quase ideal. Percebe-se também que apenas aumentar a complexidade da rede não implica em uma melhoria real da mesma.

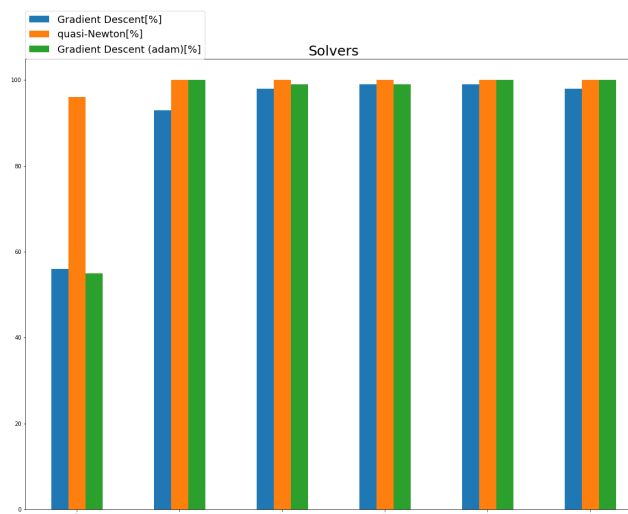
3.2. Função de Ativação

Um Perceptron pode ser modelado com diversas funções de ativação. Foi realizado ensaios com algumas funções de ativação. O gráfico de correlação das acurácias tendo como base esse parâmetro demonstra que a menos da função logística, todas as outras obtiveram um resultado melhor. O resultado inferior da função logística deve-se ao fato de não haver um tratamento prévio dos dados para que ficassem no universo de discurso dessa função.



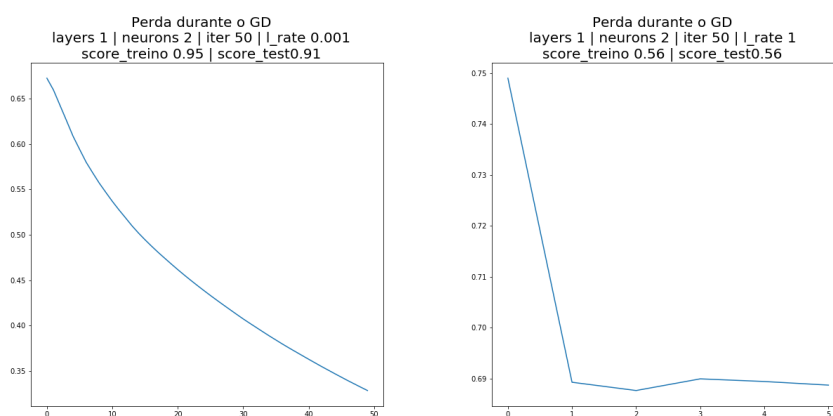
3.3. Solvers

A ideia básica de uma Rede Neural é minimizar a função de custo. Foi realizado ensaios com alguns algoritmos que realizam essa busca do mínimo da função de custo. O gráfico de correlação das acurácias tendo como base esse parâmetro, demonstra uma performance semelhante em todos com exceção do LBGFS, um algoritmo da família dos métodos quasi-Newton que aproxima o algoritmo Broyden–Fletcher–Goldfarb–Shanno usando uma quantidade limitada da memória computacional. Neste caso, percebe-se mesmo no pior caso uma performance quase ideal

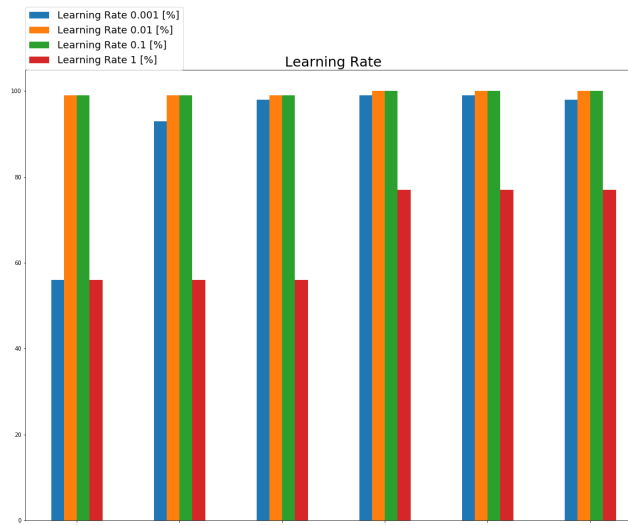


3.4. Taxa de aprendizado

Learning Rate foi outro parametro interessante que testado. Com ele percebeu-se um queda mais abrupta na função de perda do gradiente. Essa função de perda precisa-se ter um cuidado ao ser analisada assim como a fator de aprendizado, pois como pode-se concluir um aumento nessa taxa faz com que o score para o teste piore o que leva a inferir que um aumento na Taxa de Aprendizado tende a levar precocemente a um máximo local



O gráfico de correlação das acurácias tendo como base o learning rate demonstra também que para um learning rate elevado a rede acaba se perdendo e não conseguindo mais convergir



4. Considerações Finais

Devido a sua natureza, a rede neural se mostrou uma abordagem muito melhor para o problema de classificações do que o Sistema Fuzzy. A implementação (utilizando uma API) da Rede Neural foi muito mais simples e direta e a Rede Neural mostrou-se mais eficiente também no quesito de utilização de recursos computacionais.

Como sugestão para trabalho futuros, aconselha-se o uso de um especialista para a modelagem dos conjuntos fuzzy ou mesmo uma maneira otimizada de varrer o espaço de possibilidades das combinações e configurações desses conjuntos no universo de discurso. No que diz respeito a Rede Neural, devido a simplicidade do problema (variáveis linearmente independentes) a análise limitou-se em comparação de outros parâmetros no ajuste fino da rede tendo em mente que uma rede com somente 1 perceptron teria capacidade de aproximar-se do ideal na classificação.

Referências

- Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., and Varoquaux, G. (2013). API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122.
- Wang, L.-X. and Mendel, J. M. (1992). Generating fuzzy rules by learning from examples. *IEEE Transactions on systems, man, and cybernetics*, 22(6):1414–1427.
- Zimmermann, H.-J. (2012). *Practical applications of fuzzy technologies*, volume 6. Springer Science & Business Media.