

# Relatório de utilização do MLP

Gleidson Mendes Costa

May 14, 2017

## 1 Introdução

Este trabalho irá apresentar a utilização de um algoritmo para a classificação de dados não lineares.

## 2 Multilayer Perceptrons teoria

Multilayer Perceptrons(MLP) é uma estrutura constituída por nós, onde cada nó é definido como um neurônio, exceto pelos nós que alimentam a rede. Cada neurônio possui uma unidade de processamento, em que recebe como entrada os dados dos neurônios anteriores, processa a entrada e a saída é conectada ao neurônio posterior.

Cada neurônio possui um peso em cada conexão para com o próximo neurônio. E estes pesos que definem o quão necessário um valor de saída do neurônio é. Assim, consertar os pesos é necessário para permitir que a MLP consiga resultados mais precisos.

Forward propagation é o método para encontrar o valor relacionado a um determinado conjunto de características, em que os dados de entrada são submetidos à rede neural implicados pelos pesos das conexões de cada neurônio e pela função de ativação, que delimita a alteração de estado do neurônio.

Backward propagation, é o metodo para consertar os pesos, utilizado para treinar a rede neural. Importante pois de acordo com o resultado obtido no final da rede e comparado o quão distante está de obter o resultado desejado, este erro é propagado para a rede para consertar os pesos de acordo.

Bias é um neurônio de apoio, permitindo fornecer liberdade ao neurônio

## 3 MLP prática

O algoritmo desenvolvido segue os seguintes passos:

- Leitura dos objetos e respectivos rótulos
- Leitura dos parâmetros de classificação
- Definir os objetos de treino (para este caso, o treino é 100% dos dados)
- Definir os objetos de teste (para este caso, o teste é feito com 100% dos dados)

- Definir estrutura da rede neural é definir o peso das conexões de forma randômica
- Fazer treinamento da rede
- Executar forward propagation para cada objeto
- Propagar o erro com backward propagation para consertar os pesos
- Executar n vezes os conjuntos de entrada
- Para execução dos testes, realiza-se apenas forward propagation na mlp já configurada e o resultado final é feito uma aproximação para com o valor mais próximo (para este caso, 0 ou 1)

As funções de ativação usadas foram: sigmoid, tangente hiperbolica, gaussiana, linear

## 4 Experimento

O arquivo de objetos de treino e teste estão no mesmo arquivo de baterias de teste, em que nas baterias há alteração dos parâmetros relacionado ao valor do tipo de função de ativação, taxa de aprendizagem, valor de condição de para em relação ao erro:

O arquivo de objetos para treino e teste e execução é estruturado em(todos os valores devem ser numéricos):

- Linha 1: numero de objetos(R), numero de características(C), numero de classes, valor do K, numero da função de ativação(0-sigmoid, 1-tangente hiperbolica, 2-gaussiana, 3-linear), porcentagem(em decimal)
- Linhas 2 - R: Valores das características de cada objeto com o valor da classe na última coluna
- Linha R+1: numero de execuções(E), número de parâmetros
- Linha R+2 - Fim arquivo: Diferentes tipos de combinações de parâmetros(Sendo o primeiro: funcao de ativação(0-sigmoid, 1-tangente hiperbolica, 2-gaussiana, 3-linear); segundo: taxa de aprendizado; terceiro: condição de parada pelo erro.

Por praticidade a leitura do arquivo é feita por redirecionamento de entrada executando todos os testes e o resultado pode ser redirecionado:

```
.\mlp < {entrada_dataset}.{ext} > {saida}.{ext}
```

o arquivo de saída é melhor visto como arquivo .CSV separado por ";" para ser aberto em uma tabela.

Os datasets utilizados são os valores das combinações binárias de 2 digitos com relacionamento XOR.

Especificações da máquina:

Arquitetura: i386, Modo(s) operacional da CPU:32-bit, 64-bit, CPU(s): 4, Sistema Operacional: Linux( Ubuntu ), Memoria RAM: 4GB

## 5 Resultado

Depois de realizado o treinamento, em que para cada conjunto de entradas executava-se até o número máximo de épocas fosse obtido ou quando a soma dos erros quadráticos convergia para o valor predefinido nas baterias.

Logo após, executava-se um teste com todos os dados e fazia-se a aproximação. Reinicio as mlp, com novos pesos e feito o processo novamente.

Abaixo segue os resultados para diferentes tipos de combinações de valores entre funções de ativação, taxa de aprendizagem e condição de parada.

## 6 Conclusão

Esta rede primeiramente foi testada com os pesos definidos nos slides apresentados em sala de aula, tanto com os pesos para treinamento, quanto os pesos do resultado final. Foi possível obter os mesmos resultados a cada iteração. Porém, quando aplicado valores randômicos, muitas vezes demora-se muito para convergir, chegando a momentos a atingir o número máximo de épocas.

Função de sigmoid é a que melhor se comporta, enquanto as outras ou demoram para convergir ou não conseguem. Diretamente relacionado com a taxa de aprendizagem, valores pequenos convergem mais lentamente.

Assim como há problemas de executar todas as épocas permitidas, também há casos em que executa apenas 3, 4 épocas ou somente 1.

Função linear não foi possível obter resultados;

Table 1: Resultados MLP

MLP 2 2 1					
Epoques	Iteracoes	funcao ativacao	Alpha	Taxa Erro	Acurácia
500000	2000000	Sigmoid	0.1	0.01	75%
26393	105572	Sigmoid	0.1	1	100%
1504	6016	Sigmoid	0.5	0.01	100%
2345	9380	Sigmoid	0.5	1	100%
500000	2000000	Sigmoid	0.9	0.01	75%
2233	8932	Sigmoid	0.9	1	100%
672	2688	Sigmoid	2	0.01	100%
500000	2000000	TanH	2	1	50%
500000	2000000	TanH	0.1	0.01	50%
500000	2000000	TanH	0.1	1	50%
500000	2000000	TanH	0.5	0.01	50%
1	4	TanH	0.5	1	50%
500000	2000000	TanH	0.9	0.01	50%
500000	2000000	TanH	0.9	1	50%
500000	2000000	TanH	2	0.01	50%
500000	2000000	TanH	2	1	50%
23677	94708	Gaussian	0.1	0.01	100%
500000	2000000	Gaussian	0.1	1	50%
500000	2000000	Gaussian	0.5	0.01	50%
2747	10988	Gaussian	0.5	1	100%
500000	2000000	Gaussian	0.9	0.01	75%
500000	2000000	Gaussian	0.9	1	75%
500000	2000000	Gaussian	2	0.01	75%
6456	25824	Gaussian	2	1	100%
2	8	Linear	0.1	0.01	50%
2	8	Linear	0.1	1	50%
2	8	Linear	0.5	0.01	50%
2	8	Linear	0.5	1	50%
2	8	Linear	0.9	0.01	50%
2	8	Linear	0.9	1	50%
2	8	Linear	2	0.01	50%
2	8	Linear	2	1	50%