

Relatório 05

Redes Neurais Artificiais

O Perceptron

Cristiano Lopes Moreira

Matrícula: 119103-0

Aluno		RA/Matrícula		Professor		Tipo	
Cristiano Lopes Moreira		119103-0		Dr Reinaldo Bianchi		Relatório de implementação	
Data	Versão	Turma		Nome do arquivo			Página
27/11/2019	1	2º. Semestre de 2019		PEL_208_Relatório_05_Cristiano_Moreira.doc			1 (14)

Sumário

1.	Introdução	3
2.	Desenvolvimento teórico	3
3.	Proposta de implementação Algoritmo Perceptron	6
4.	Experimentação e Resultados	8
4.1.	Dados Porta Logica E	9
4.2.	Dados Porta Logica OU	10
4.3.	Dados Porta Logica XOR	11
4.4.	Iris Fisher	12
5.	Conclusão	14
6.	Referências	14

Aluno		RA/Matrícula	Professor	Tipo	
Cristiano Lopes Moreira		119103-0	Dr Reinaldo Bianchi	Relatório de implementação	
Data	Versão	Turma	Nome do arquivo		Página
27/11/2019	1	2º. Semestre de 2019	PEL_208_Relatório_05_Cristiano_Moreira.doc		2 (14)

1. Introdução

Com aplicações nas áreas de reconhecimento e classificação de padrões de escrita e fala, reconhecimento de faces em visão computacional, controle previsão de ações no mercado financeiro, identificação de anomalias em imagens médicas, entre outras; as redes neurais artificiais (RNA) vieram da ideia de se construir uma máquina ou mecanismo autônomo, que seja dotado de inteligência, inspirada na visão do cérebro humano e dos neurônios biológicos que tem grande poder de processamento paralelo, capacidade armazenamento e de utilizar conhecimento experimental. Essas redes são constituídas de sub elementos, denominados neurônios, que realizam, de forma paralelizada, diversas subtarefas de um problema maior (HAYKIN, 2008).

Dentre os modelos de RNA, a rede Perceptron, idealizada por Rosenblatt (1958), é uma arquitetura de aprendizado supervisionado, onde a rede aprende com uma saída já conhecida, para classificação binária de padrões que se limitam a problemas linearmente separáveis, obtendo saída discreta tipo verdadeiro ou falso.

A saída da rede é tratada pela ponderação ponto a ponto entre os padrões de entradas e seus pesos sinápticos, a fim de serem ativados pela soma ponderada destes estímulos que o neurônio aplicará na fronteira de decisão com uma função de ativação. O objetivo desta rede é encontrar os melhores parâmetros, ou pesos, através do ajuste fino de seus hiperparâmetros (taxa de aprendizado, número de épocas, função de ativação, quantidade de padrões...) a fim de encontrar o erro zero entre as saídas da rede e a desejada conseguindo a separabilidade das classes limitado pelo hiperplano.

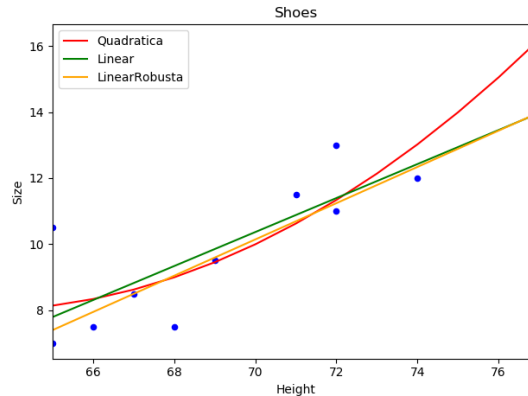
2. Desenvolvimento teórico

A regressão linear é um dos algoritmos mais conhecidos e utilizados em estatística e em aprendizado de máquina, sua representação linear é uma equação matemática que melhor descreve uma reta que se encaixa entre os pontos

Aluno		RA/Matrícula	Professor	Tipo	
Cristiano Lopes Moreira		119103-0	Dr Reinaldo Bianchi	Relatório de implementação	
Data	Versão	Turma	Nome do arquivo		Página
27/11/2019	1	2º. Semestre de 2019	PEL_208_Relatório_05_Cristiano_Moreira.doc		3 (14)

amostrados de uma variável X com saída representando a descrição de um fenômeno.

$$y = f(X) = \alpha + \beta \quad (1)$$



Suponha que Y seja conhecido até um número finito de pontos p dos parâmetros $\beta = (\beta_1, \dots, \beta_p)$, ou seja, $Y = X\beta$. Estimamos β pelo valor de coeficiente $\hat{\beta}$ que melhor se ajusta aos dados.

A rede neural artificial perceptron, assim como o método de regressão linear, busca encontrar os coeficientes da reta que melhor representa estes pontos, porém, distinto do método de regressão linear pelos mínimos quadrados, o coeficiente de estimativa calculado pela RNA perceptron é obtido através de diversas interações, chamadas de treinamento ou aprendizado, que minimizam o erro do coeficiente com base na confirmação do resultado esperado balizado por um coeficiente de aprendizado.

Matematicamente os coeficientes da reta, ou pesos sinápticos, são vetorialmente representados por " w ", com seu aprendizado/atualização, realizado pelas diversas interações nos ciclos de aprendizado (episódios ou épocas) com as equações abaixo:

$$w_i = w_i + \Delta w_i \quad (2)$$

Aluno Cristiano Lopes Moreira		RA/Matrícula 119103-0	Professor Dr Reinaldo Bianchi	Tipo Relatório de implementação	
Data 27/11/2019	Versão 1	Turma 2º. Semestre de 2019	Nome do arquivo PEL_208_Relatório_05_Cristiano_Moreira.doc		Página 4 (14)

Sendo: $\Delta w_i = \eta(t - o)x_i$ (3)

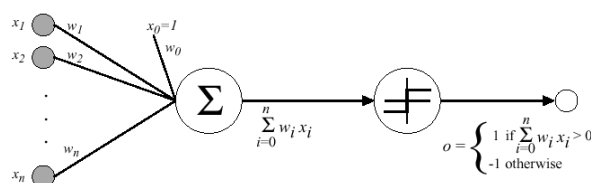
Onde: t é o valor desejado (4)

o é a saída do perceptron

x_i é a entrada do perceptron

η é a taxa de aprendizado

Uma vez concluído o processo de treinamento, o perceptron pode ser utilizado para estimar casos que não fazem parte do conjunto de dados do treinamento. Através da somatória de todas as variáveis de entrada, balizadas pelos pesos sinápticos, o perceptron verifica por função de ativação se as entradas alcançam seu nível de saturação para acionar a saída do perceptron com 1 ou -1 (classificação binária), que ao final é a estimativa de qual classe pertence o item pesquisado.



Função de ativação (5)

$$f(W^T X) = \begin{cases} 1 & \text{se } \sum_{i=0}^n w_i x_i > 0 \\ -1 & \text{caso contrário} \end{cases}$$

Parâmetro	Variável Representativa	Tipo Característico
Entradas	X_i (i-ésima entrada)	Reais ou binários
Pesos Sinápticos	w_i (associado a x_i)	Reais iniciados aleatoriamente
Saída	y	Binária
Função de Ativação	$g(x)$	Degrau bipolar
Processo de Treinamento	-----	Supervisionado
Regra de Aprendizado	-----	Regra de Hebb

Aluno		RA/Matrícula		Professor		Tipo	
Cristiano Lopes Moreira		119103-0		Dr Reinaldo Bianchi		Relatório de implementação	
Data	Versão	Turma		Nome do arquivo			Página
27/11/2019	1	2º. Semestre de 2019		PEL_208_Relatório_05_Cristiano_Moreira.doc			5 (14)

3. Proposta de implementação Algoritmo Perceptron

Pseudocódigos:**Treinamento**

Obtém-se um conjunto de dados;

Associar a saída desejada para cada amostra obtida;

Iniciar um vetor w com valores aleatórios pequenos;

Especificar uma taxa de aprendizagem η_{aprend}

definir um número de episódios: $\text{epsodios} \leftarrow \text{episódios de aprendizagem}$

Para $i = 1$ até epsodios faça:

$\text{erro} \leftarrow \text{"inexistente"}$

 para todas as amostras de treinamento faça:

$u \leftarrow W^T X^{(k)}$

$y \leftarrow \text{sinal}(u)$

 se $y \neq$ saída associada desejada (S)

$w \leftarrow w + \eta_{\text{aprend}} (S - y) X^{(k)}$

$\text{erro} += 1$

Aluno		RA/Matrícula		Professor		Tipo	
Cristiano Lopes Moreira		119103-0		Dr Reinaldo Bianchi		Relatório de implementação	
Data	Versão	Turma		Nome do arquivo			Página
27/11/2019	1	2º. Semestre de 2019		PEL_208_Relatório_05_Cristiano_Moreira.doc			6 (14)

Predição

Obtém-se a amostra a ser classificada X ;

Utilizar o vetor w ajustando durante o treinamento;

Para $i = 1$ até episódios faça:

$u \leftarrow W^T X$

$y \leftarrow \text{sinal}(u)$

se $y = -1$

retorna Classe 0

se não

retorna Classe 1

Aluno		RA/Matrícula		Professor		Tipo	
Cristiano Lopes Moreira		119103-0		Dr Reinaldo Bianchi		Relatório de implementação	
Data	Versão	Turma		Nome do arquivo			Página
27/11/2019	1	2º. Semestre de 2019		PEL_208_Relatório_05_Cristiano_Moreira.doc			7 (14)

4. Experimentação e Resultados

Para verificar o funcionamento do algoritmo do Perceptron, foi realizada a implementação em Python confrontando os resultados entre a classificação indicada na base de dados e o agrupamento proposto pelo algoritmo:

Ambiente:

PyCharm 2019.2.2 (Professional Edition) Build#PY-192.6603.34

Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 01:31:54) on win32

Bibliotecas:

matplotlib-3.1.1	(utilizado para plotagem de gráficos)
pandas-0.25.2	(suporte à plotagem de gráficos)
xlrd-1.2.0	(leitura de arquivos do Excel - base de dados)
numpy-1.17.4	(gestão de matrizes)

Base de Dados:

dbTraining.xlsx	E	Porta logica E de 2 entradas
	OU	Porta logica OU de 2 entradas
	XOR	Porta logica XOR de 2 entradas
	Íris	http://archive.ics.uci.edu/ml/datasets/Iris

Aluno		RA/Matrícula	Professor	Tipo	
Cristiano Lopes Moreira		119103-0	Dr Reinaldo Bianchi	Relatório de implementação	
Data	Versão	Turma	Nome do arquivo		Página
27/11/2019	1	2º. Semestre de 2019	PEL_208_Relatório_05_Cristiano_Moreira.doc		8 (14)

4.1. Dados Porta Lógica E

Base de dados:

X	Y	Classificação	
		Original	Perceptron
0	0	0	0
0	1	0	0
1	0	0	0
1	1	1	1

Perceptron Porta E			
Função de ativação: $f(W^T X) = \begin{cases} 1 & \text{se } \sum_{i=0}^n w_i x_i \geq \alpha \\ -1 & \text{caso contrário} \end{cases}$			
Dimensões	d	2	
Taxa de aprendizagem	η	0.01	
Coeficiente de saturação	α	0.5	
Episódios/eras	E	38	
Pesos Sinápticos	W	(0.5577 0.5796 -0.0855)	

Resultados:

		Classificação Correta		Precisão
		0	1	
Perceptron	0	3	0	100%
	1	0	1	100%
Precisão Total				100%

Os testes demonstraram que com 1 perceptron é possível representar com 100% de precisão uma porta lógica do tipo E

Aluno		RA/Matrícula		Professor		Tipo	
Cristiano Lopes Moreira		119103-0		Dr Reinaldo Bianchi		Relatório de implementação	
Data	Versão	Turma		Nome do arquivo			Página
27/11/2019	1	2º. Semestre de 2019		PEL_208_Relatório_05_Cristiano_Moreira.doc			9 (14)

4.2. Dados Porta Lógica OU

Base de dados:

X	Y	Classificação	
		Original	Perceptron
0	0	0	0
0	1	1	1
1	0	1	1
1	1	1	1

Perceptron Porta OU			
Função de ativação: $f(W^T X) = \begin{cases} 1 & \text{se } \sum_{i=0}^n w_i x_i \geq \alpha \\ -1 & \text{caso contrário} \end{cases}$			
Dimensões	d	2	
Taxa de aprendizagem	η	0.01	
Coeficiente de saturação	α	0.5	
Episódios/eras	E	58	
Pesos Sinápticos	W	(0.083 0.4356 0.4231)	

Resultados:

		Classificação Correta		Precisão
		0	1	
Perceptron	0	1	0	100%
	1	0	3	100%
Precisão Total				100%

Os testes demonstraram que com 1 perceptron é possível representar com 100% de precisão uma porta lógica do tipo OU

Aluno		RA/Matrícula		Professor		Tipo	
Cristiano Lopes Moreira		119103-0		Dr Reinaldo Bianchi		Relatório de implementação	
Data	Versão	Turma		Nome do arquivo		Página	
27/11/2019	1	2º. Semestre de 2019		PEL_208_Relatório_05_Cristiano_Moreira.doc		10 (14)	

4.3. Dados Porta Lógica XOR

Base de dados:

X	Y	Classificação	
		Original	Perceptron
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	0

Perceptron Porta XOR		
Função de ativação: $f(W^T X) = \begin{cases} 1 & \text{se } \sum_{i=0}^n w_i x_i \geq \alpha \\ -1 & \text{caso contrário} \end{cases}$		
Dimensões	d	2
Taxa de aprendizagem	η	0.01
Coeficiente de saturação	α	0.5
Episódios/eras	E	200
Pesos Sinápticos	W	(0.5577 0.5796 -0.0855)

Resultados:

		Classificação Correta		Precisão
		0	1	
Perceptron	0	1	2	50%
	1	1	0	0%
Precisão Total				25%

Os testes demonstraram que com 1 perceptron NÃO é possível representar uma porta lógica do tipo XOR, isto ocorre porque o perceptron classifica somente massas linearmente divisíveis, o que não é o caso da porta lógica XOR.

Aluno		RA/Matrícula	Professor	Tipo	
Cristiano Lopes Moreira		119103-0	Dr Reinaldo Bianchi	Relatório de implementação	
Data	Versão	Turma	Nome do arquivo		Página
27/11/2019	1	2º. Semestre de 2019	PEL_208_Relatório_05_Cristiano_Moreira.doc		11 (14)

4.4. Iris Fisher

Base de dados:

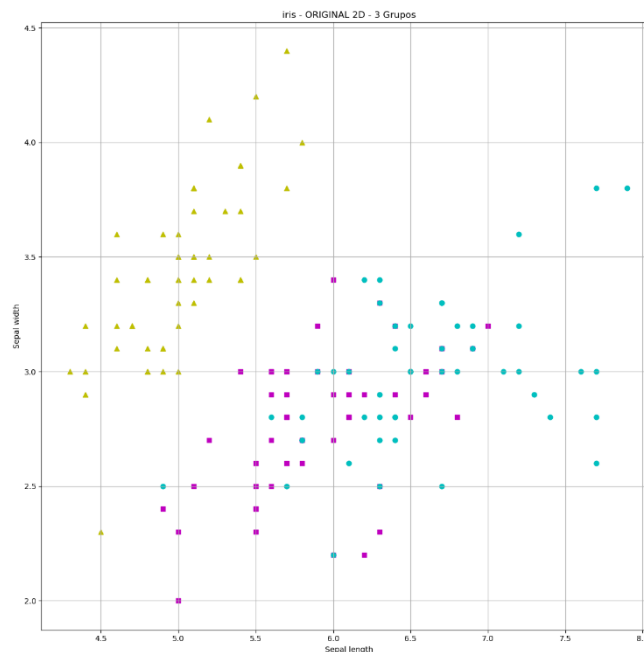


Fig2: Distribuição Original 3 dimensões visão em 2D da base de dados Iris-Fisher

O conjunto de dados contém 3 classes de 50 instâncias cada, em que cada classe se refere a um tipo de planta de íris, mostrados na Figura 2.

Uma classe é linearmente separável das outras duas e as demais não são linearmente separáveis.

Atributo previsto: classe da planta íris.

Informações da base:

- comprimento da sépala em cm
- largura da sépala em cm
- comprimento da pétala em cm
- largura da pétala em cm
- 3 classes: (Setosa, Versicolour e Virginica)

Aluno		RA/Matrícula		Professor		Tipo	
Cristiano Lopes Moreira		119103-0		Dr Reinaldo Bianchi		Relatório de implementação	
Data	Versão	Turma		Nome do arquivo			Página
27/11/2019	1	2º. Semestre de 2019		PEL_208_Relatório_05_Cristiano_Moreira.doc			12 (14)

Perceptron Base de Dados Iris				
Função de ativação: $f(W^T X) = \begin{cases} 1 & \text{se } \sum_{i=0}^n w_i x_i \geq \alpha \\ -1 & \text{caso contrário} \end{cases}$				
Dimensões	d	4		
Taxa de aprendizagem	η	0.01		
Coefficiente de saturação	α	0.5		
Episódios/eras	E	200		
Pesos Sinápticos	W	(-0.0787 -0.0899 0.3183 0.3075 0.5158)		

Resultados:

		Classificação Correta			Precisão
		0	1	2	
Perceptron	0	47	0	0	94%
	1	3	50	50	100%
	2	0	0	0	0%
Precisão Total					64.66%

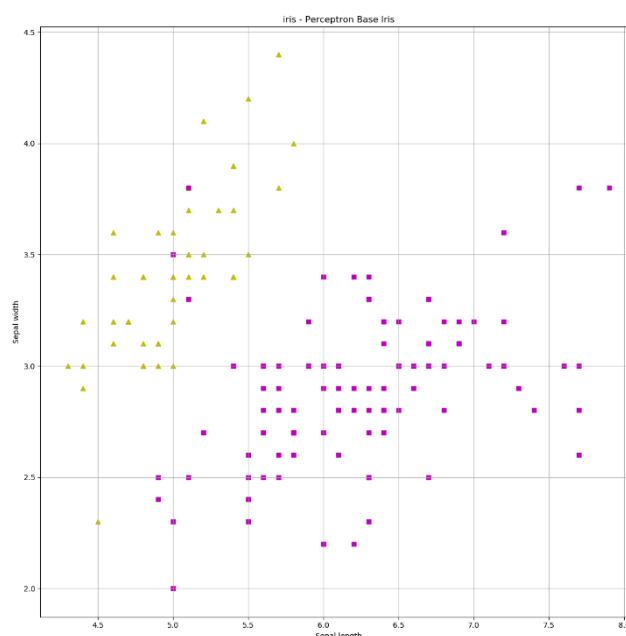


Fig3: Classificação perceptron da base de dados Iris-Fisher – diversas dimensões – vista 2D

Aluno		RA/Matrícula	Professor	Tipo	
Cristiano Lopes Moreira		119103-0	Dr Reinaldo Bianchi	Relatório de implementação	
Data	Versão	Turma	Nome do arquivo		Página
27/11/2019	1	2º. Semestre de 2019	PEL_208_Relatório_05_Cristiano_Moreira.doc		13 (14)

Pela figura 3 e pela tabela de confusão observar-se que o perceptron é eficiente para classificação dos dados que estão linearmente separados a classificação entre a Iris do grupo 0 (Setosa) e as dos grupos 1 (Versicolour) e 2 (Virginica), porém, por ser um classificador binário linear, não consegue realizar uma classificação entre os grupos 1 e 2 que não são linearmente separáveis.

O algoritmo apresentou 64.66% de acerto na classificação, em comparação com a classificação real informada pela base de dados, sendo o erro maior entre os grupos 1 e 2, onde toda Virginica foi classificada como Versicolour, sendo 100% de erro na classificação das Virginicas.

5. Conclusão

Conclui-se que o Perceptron é um algoritmo de redes neurais supervisionado de camada única pertencente a arquitetura sem realimentação (feedforward) que tem por objetivo classificar grupos linearmente separáveis, ou seja, é um classificador binário, muito útil para problemas de classificação binária e linear. É ineficiente para problemas não lineares e classificações de múltiplos grupos, porém pode ser utilizado como base de uma rede de múltiplas camadas de perceptron que deve ser objeto de um outro estudo.

6. Referências

- [1] HAYKIN, Simon. **Redes Neurais: princípios e práticas**. 2. ed. Ontario Canada: Bookman, 2008.

Aluno		RA/Matrícula	Professor	Tipo	
Cristiano Lopes Moreira		119103-0	Dr Reinaldo Bianchi	Relatório de implementação	
Data	Versão	Turma	Nome do arquivo		Página
27/11/2019	1	2º. Semestre de 2019	PEL_208_Relatório_05_Cristiano_Moreira.doc		14 (14)