



UNIVERSIDADE ESTADUAL DO MARANHÃO-UEMA
CENTRO DE CIÊNCIAS TECNOLÓGICAS - CCT
DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO
CURSO DE ENGENHARIA DE COMPUTAÇÃO
DISCIPLINA: TÓPICOS ESPECIAIS EM PROGRAMAÇÃO

Bruno Rocha Gomes

REDES NEURAIIS

SÃO LUÍS - MA

2017

Bruno Rocha Gomes

REDES NEURAIIS

Relatório apresentado a Universidade Estadual do Maranhão como requisito para obtenção de nota na disciplina Tópicos Especiais em Programação ministrada pelo professor Thiago Lemos.

SÃO LUÍS - MA

2017

RESUMO

Com o avanço da tecnologia, foram surgindo formas de ensinar máquinas a desenvolverem certas ações, através da programação. Estes métodos consistem no aprendizado de máquina, o qual uma de suas técnicas consiste em redes neurais: métodos de aprendizado com base em experiências, que se assemelham ao funcionamento dos neurônios presentes no cérebro humano. Pensando nisso, foi desenvolvida uma rede neural, cujo objetivo era obter valores o mais próximo possível da função $\cos(2x)$. O programa teria uma base de dados, onde seriam feitos os treinos, e em seguida, uma base de testes para a verificação da eficiência do mesmo. Por fim, são gerados gráficos que comparam as funções referentes ao $\cos(2x)$ e aos valores obtidos pela rede neural.

Palavras-Chave: Rede Neural; Aprendizado de Máquina; Funções.

SUMÁRIO

1. INTRODUÇÃO	1
2. OBJETIVOS.....	1
3. RESULTADOS E DISCUSSÕES	1
4. CONCLUSÃO	3
REFERÊNCIAS	4

1. INTRODUÇÃO

Entende-se por rede neural, como um conjunto de técnicas computacionais existentes para o aprendizado de máquina e que apresentam um modelo matemático baseado na estrutura dos neurônios presentes nos organismos, que são capazes de adquirir conhecimento através da experiência. Uma vez que o cérebro de um mamífero pode conter bilhões de neurônios e que estes têm a capacidade de se comunicar através das sinapses, é possível afirmar que no cérebro humano são formadas redes muito complexas, já que nele existem em média mil a dez mil sinapses para cada neurônio. Dessa forma, a técnica de rede neural busca uma forma de aprendizado semelhante, com base nos treinos realizados.

Uma rede neural é composta por várias unidades de processamento, onde cada uma se encontra conectada por canais de comunicação que possuem um determinado peso. Cada unidade realiza suas operações somente sobre seus dados recebidos como valores de entrada de suas conexões. Dessa forma, as interações que ocorrem entre as unidades de processamento da rede são o fator responsável pelo comportamento inteligente de uma Rede Neural.

2. OBJETIVOS

O objetivo deste trabalho é construir uma rede neural capaz de obter valores próximos da função $\cos(2x)$. Para o seu desenvolvimento foi utilizada a linguagem de programação Python na sua versão 3.6 e a IDE Pycharm, sendo necessária a instalação da biblioteca Pybrain3 para o uso da rede neural. Após o desenvolvimento da rede, objetiva-se construir um gráfico que mostre a função referente à $\cos(2x)$ e à rede, comparando assim, o resultado obtido pelo programa. Dessa forma, quanto mais próximo o gráfico da rede neural estiver do gráfico de $\cos(2x)$, mais eficiente é a rede desenvolvida.

3. RESULTADOS E DISCUSSÕES

Uma vez que a biblioteca Pybrain3 se encontra instalada e pronta para o uso, são declaradas funções para o desenvolvimento da rede neural. Abaixo, é mostrado um trecho do código em que é possível ver na linha 15 do código a função “SupervisedDataSet(1, 1)”, que indica que a rede trabalhará com apenas um valor de entrada e um de saída. Na linha 23, está a função “addSample(x, math.cos(2*x))”, que informa que a rede neural realizará treinos onde o valor de entrada é a variável “x” e o valor de saída é a função “ $\cos(2x)$ ”. O treino é realizado no intervalo de -4 a 4 com um passo de 0,05, o que está indicado pelo laço while.

```
15     ds = SupervisedDataSet(1, 1)
16
17     valoresX = []
18     valoresY = []
19     rede = []
20
21     x = -4
22     while(x <= 4):
23         ds.addSample(x, math.cos(2*x))
24         x += 0.05
```

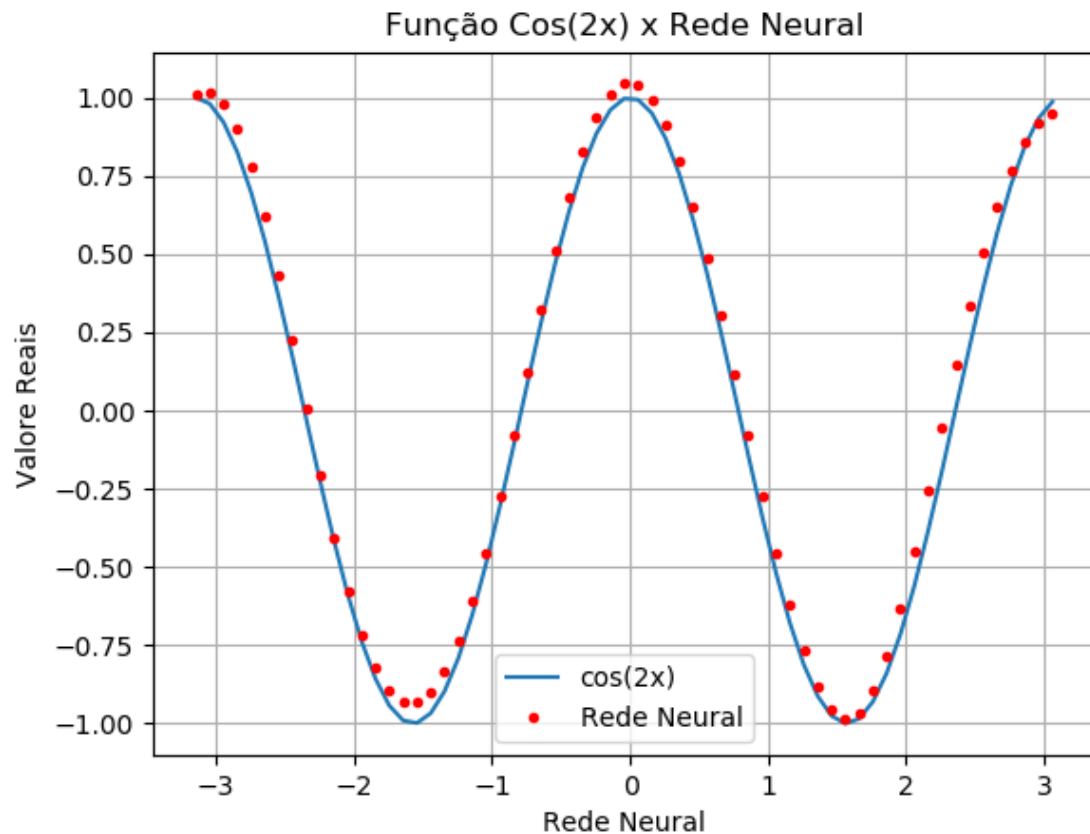
Para a criação da rede propriamente dita, é utilizada a função “buildNetwork(1, 100, 1, bias=True)”, presente na linha 27 e mostrada na imagem abaixo. Estes parâmetros, da esquerda para a direita respectivamente, significam que existirá apenas um neurônio de entrada na rede; que a rede possuirá 100 neurônios na camada oculta; que existirá apenas um neurônio de saída; e que a rede neural fará uso de “bias”, um neurônio especial capaz de aumentar os graus de liberdade da rede, além de uma melhor adaptação ao seu objetivo. Na linha 29 do código, a função “BackpropTrainer(neuralNet, ds)” indica qual a rede será utilizada, seguida do sua respectiva base de treinos como parâmetro. A função responsável por realizar os treinos da rede neural é “trainer.train()”, na linha 32, que é repetida em 2000 épocas, de acordo com o laço for. Por fim, a função “neuralNet.activate([x])”, linha 37, indica que parâmetro será utilizado para os testes da rede, o que no caso, foi o valor de “x” em um intervalo de $-\pi$ a π , com um passo de 0,1, conforme o laço de repetição while.

```
27     neuralNet = buildNetwork(1, 100, 1, bias=True)
28
29     trainer = BackpropTrainer(neuralNet, ds)
30
31     for i in range(2000):
32         print("Erro: ", trainer.train())
33
34
35     x = -1*math.pi
36     while (x < 1*math.pi):
37         valoresY.append(math.cos(2*x))
38         rede.append(neuralNet.activate([x]))
39         valoresX.append(x)
40         x += 0.1
```

A função “trainer.train()” exibe um erro a cada iteração, o qual vai diminuindo à medida que a rede vai aprendendo o que lhe é determinado na base de treinos. Para a rede desenvolvida, o erro encontrado decai até aproximadamente “0,001”, conforme mostra a imagem abaixo.

```
Erro: 0.00161510477105
Erro: 0.00161931883524
Erro: 0.00164240714742
Erro: 0.00162154890808
Erro: 0.00168488029096
Erro: 0.00155098707857
Erro: 0.00166219766021
Erro: 0.00168006593483
Erro: 0.00175011480427
```

Uma vez que o erro se encontra bem pequeno, é de se esperar que a rede tenha alcançado certa eficiência, estando o seu valor então, bem próximo do seu objetivo, que é a função “ $\cos(2x)$ ”. Sendo assim, para uma melhor visualização da rede em relação ao seu objetivo, é plotado um gráfico indicando o mesmo.



Com o gráfico foi possível observar melhor a aproximação da rede neural desenvolvida à função “ $\cos(2x)$ ”. Dessa forma, a base de dados utilizada, assim como os intervalos e épocas de treino, foi suficiente para a elaboração de uma rede eficiente.

4. CONCLUSÃO

A técnica de rede neural é uma das mais utilizadas para o aprendizado de máquina, no ramo da Inteligência Artificial. Esta técnica é baseada na capacidade de aprendizado e raciocínio dos neurônios presente nos cérebros do ser humano.

Para o desenvolvimento de uma rede neural que atingisse valores próximos ao da função $\cos(2x)$, foi observado que com o uso da biblioteca Pybrain, foi possível criar facilmente uma rede neural com estas características. Além disso, o gráfico gerado referente à mesma foi capaz de comprovar a eficiência da rede, uma vez que os valores obtidos foram bastante satisfatórios.

REFERÊNCIAS

- 1- ICMC – USP. **Redes Neurais Artificiais**. Acessado em Dezembro de 2017.
Disponível em: < <http://conteudo.icmc.usp.br/pessoas/andre/research/neural/>>
- 2- Din – Departamento de Informática (UEM). **Redes Neurais**. Acessado em Dezembro de 2017. Disponível em: < <http://www.din.uem.br/ia/neurais/>>
- 3- **Redes neurais com python usando o pybrain**. Acessado em Dezembro de 2017.
Disponível em: < <https://www.youtube.com/watch?v=Fba4UUqupiA>>