INSTITUTO FEDERAL DO SUL DE MINAS GERAIS
Campus Poços de Caldas

Método de Newton para o Aprendizado do Neurônio Artificial

Joaquim Augusto Pomarico[1], Marcos Vinícius Moreira[2], Otávio Messias Palma[3], Pablo Borges dos Santos[4], Yuri Fernandes[5],
joaquim@alunos.ifsuldeminas.edu.br[1], marcos@alunos.ifsuldeminas.edu.br[2],
otavio.palma@alunos.ifsuldeminas.edu.br[3],
pablo.borges@alunos.ifsuldeminas.edu.br[4], yuri@alunos.ifsuldeminas.edu.br[5]
Poços de Caldas, 09 de Dezembro de 2019

Resumo: *Para uma rede neural complexa, com parâmetros muito grandes, a tarefa de encontrar o mínimo dentre vários vales contido em tal superfície é muito difícil e custosa. Conforme visto na disciplina, um dos métodos para encontrar tal mínimo é a utilização do Método Gradiente Descendente. Todavia, para redes neurais grandes e com mais de uma camada oculta, o método gradiente se torna menos eficiente. Um dos problemas para uso do método gradiente em grandes redes é o fato do mesmo utilizar método de otimização de primeira ordem, assumindo sempre que a superfície da rede é um plano, sem contar com curvaturas. Uma das soluções para os pontos negativos do método gradiente é aplicação do método de Newton. O método de Newton é um método iterativo de aproximação, geralmente utilizado em problemas de otimização de segunda ordem e para aproximação de funções.*

1 Introdução

- Utilização do Método de Newton para aprendizado do Neurônio Artifical ao invés do Método do Gradiente visto em sala.
- O algoritimo utilizado foi o RNA desenvolvido em sala junto do professor da disciplina, sendo que foram feitas alterações na evolução dos neurônios com a implementação do Método de Newton.
- Foi utilizada a lingugaem de programação Java e as IDEs VSCode e NetBeans.
- Breve descrição dos resultados.

2 Fundamentação teórica

- Em análise numérica, o método de Newton, desenvolvido por Isaac Newton e Joseph Raphson, tem como objetivo estimar raízes de uma função, para isto escolhe-se uma aproximação inicial para esta e calcula-se a equação da reta tangente (através de

uma derivada) da função neste ponto e a interseção dela com o eixo das abcissas, a fim de se encontrar uma melhor aproximação para a raíz. Repetindo-se o processo, cria-se um método iterativo para encontrarmos a raíz da função.

- Fórmula do Método de Newton:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, n \in \mathbb{N} \tag{1}$$

Onde o valor inicial de x é estimado no começo da iteração.

3 Algoritmo

- O algoritmo é fundamentalmente o mesmo implementado em aulas com algumas pequenas alterações.
- A função de ativação definida para uso foi a Limiar segundo conselho do professor. A rede neural artificial escolhida foi o Perceptron, novamente sob conselho do professor, dessa forma, não é possível lidar com mais de 2 camadas.
- A Rede Neural Artifical Perceptron é inicializada com o número de entradas adequado ao caso e tem a função de ativação configurada junto do número de épocas e dos vetores de aprendizado (entrada e saída esperada).
- Durante o aprendizado, atualiza-se os pesos da entradas a partir do valor inicial estimado, que no caso do algoritmo é gerado de forma aleatória entre 0,0 e 1,0. A cada iteração nova, calcula-se o erro até que este seja zerado, o que signifca que o valor desejado foi encontrado.
- Utilizando o método de Newton, diferente do Método do Gradiente, temos a possibilidade de estar dividindo por 0, dessa forma trata-se esse caso somente replicando o valor da entrada para a próxima iteração.
- No final da iteração mostra-se o erro total da epoca, caso seja 0 a execução foi bem sucedida, caso contrário não.
- No caso de uma execução bem sucedida o valor de saída indicado na linha abaixo será de acordo com os dados de entrada.
- A ideia de usar o algoritmo pronto veio da própria sugestão do professor que nos orientou a somente fazer as mudanças no código ao invés de refatorá-lo inteiro. Pode-se aproveitar do código pronto e focar nas alterações necessárias com mais afinco visto que não precisamos nos preocupar com a funcionalidade do código que ja fora definida como confiável previamente.

4 Recursos de implementação

- Descrição das tecnologias usadas para a implementação do algoritmo.
- Se alguma biblioteca externa for usada, então esta biblioteca deverá ser devidamente citada. Os recursos da biblioteca que forem usados deverão estar detalhados nesta seção.
- Se para o contexto prático foi preciso criar ou usar alguma base de dados, então o grupo deverá explicar a base e os dados que compõe a base.

5 Código fonte

- Código fonte deverá ser enviado junto com este relatório.
- Não enviar o programa compilado.
- Descrever como compilar e executar o programa, se necessário.
- Apresentar os aspectos mais importantes da implementação. Veja o exemplo a seguir.

```
1 // Exemplo hipotético de classe.
2 public class Main {
3
4     public static void main(String args[]) {
5
6         // Cria uma Rede Neural Artificial
7         RNA redeNeural = new RNA();
8
9         // Treina a Rede Neural.
10        redeNeural.train();
11
12    }
13
14
15 }
```

6 Conclusão

- Descrever os resultados obtidos com a aplicação do algoritmo sobre o contexto.
- Descrever quaisquer limitações encontradas.
- Descrever a contribuição do trabalho para a construção do conhecimento do grupo.

Observações: Serão avaliados: o relatório do trabalho prático, o código fonte e a aplicação do algoritmo sobre o contexto prático.

$$NF = (NC + NR + NA) * 0.1, \tag{2}$$

onde:

- NF: é a nota final do trabalho.
- NR: é a nota sobre o relatório.
- NC: é a nota da implementação, do código fonte do algoritmo e dos seus resultados sobre o contexto prático.
- NA: é a nota da apresentação e arguição oral do trabalho.
- $NC \in [0,10], NR \in [0,10], NA \in [0,10]$ e $NF \in [0,3]$.