**Universidade Federal de Pernambuco - UFPE**

**CIn - Centro de Informática**

**REDES NEURAIS**

**Mini Projeto 2**

Antonio José Gadelha de Albuquerque Neto(ajgan)

Rielson Leandro da Silva de Lima(rlsl)

**Recife, 2018.2**

Os resultados foram obtidos usando o roteiro do link fornecido no site da cadeira. Assim como no mini projeto 1, o estudo disponível no link estava bem detalhado e o código estava disponível

O projeto foi rodado em Python 2.7 pois alguns comandos utilizados no código original não funcionam em Python 3. Ao invés de mudarmos o código original, decidimos por executar utilizando Python 2 mesmo. Algumas alterações no código-fonte precisaram ser feitas pois algumas dependências tinham mudado, as alterações não alteraram o funcionamento do programa.

Uma das alterações feitas, foi desconsiderar um passo feito pelo desenvolvedor, que só avaliava o conjunto de teste caso a acurácia no conjunto de validação aumentasse. Entendemos a razão desse passo, mas consideramos desnecessário para o objetivo desse mini projeto

O código foi executado baseado em duas abordagens, uma de uma rede simples e outra com uma rede com 3 camadas, assim como foi feito pelo desenvolvedor. Ao adicionar mais camadas, o processo de treinamento e teste ficou muito lento, gastando horas para avaliar coisas que foram avaliadas em poucos minutos com a rede simples. Por falta de tempo, optamos por reduzir o número de épocas para a rede com mais camadas(de 60 para 20).

O código e os resultados podem ser observados no Github através do link: <https://github.com/ajgan/RN-Projeto2>

Aqui segue um print do terminal evidenciando a execução para todas as instâncias de teste e um outro print mostrando os resultados apenas para a rede com mais camadas. Temos prints da execução de todas as classes nas duas redes, mas achamos que não seria interessante colocar tudo no relatório, já que exibiremos uma tabela com os resultados condensados. As demais imagens podem ser verificadas no link do Github já fornecido anteriormente



*Execução para todas as classes na rede com uma camada*



*Execução para todas as classes na rede com mais camadas*

As épocas começam a contagem no 0. O total de épocas observadas foi 60 para cada classe na rede simples e 20 para a rede com mais camadas. Vale lembrar que como a inicialização dos pesos é randômica, os resultados não seriam iguais em outras execuções.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test**  **Data** | **Melhor Época**  **Rede Simples** | **Melhor Resultado**  **Rede Simples** | **Melhor Época**  **Rede Com 3 Camadas** | **Melhor Resultado**  **Rede Com 3 Camadas** |
| Tudo | 59 de 60 | 97,67% | 19 de 20 | **98,92%** |
| Classe 0 | 36 de 60 | 98,88% | 20 de 20 | **98,98%** |
| Classe 1 | 28 de 60 | **99,29%** | 10 de 20 | 99,03% |
| Classe 2 | 24 de 60 | 97,77% | 20 de 20 | **99,32%** |
| Classe 3 | 17 de 60 | 98,81% | 20 de 20 | **99,50%** |
| Classe 4 | 60 de 60 | **97,96%** | 20 de 20 | 97,45% |
| Classe 5 | 14 de 60 | 97,53% | 05 de 20 | **98,65%** |
| Classe 6 | 27 de 60 | 98,42% | 19 de 20 | **99,16%** |
| Classe 7 | 48 de 60 | 97,75% | 06 de 20 | **99,31%** |
| Classe 8 | 52 de 60 | 97,32% | 20 de 20 | **98,66%** |
| Classe 9 | 55 de 60 | 97,20% | 20 de 20 | **97,30%** |

Os resultados observados foram bem satisfatórios. Percebemos que mesmo a rede básica obteve melhores resultados que o do mini projeto 1. Foi notável perceber também que a rede com mais camadas obteve, em geral, resultados mais precisos mesmo sendo executada com um terço das épocas da rede simples(devido à complexidade do tempo). Entendemos que conseguiríamos obter resultados ainda melhores, colocando mais épocas e talvez até mais camadas, porém não seria possível devido ao tempo. Essas execuções foram boas para criarmos um bom senso do tempo necessário para o grande projeto.