**Universidade Federal de Pernambuco - UFPE**

**CIn - Centro de Informática**

**REDES NEURAIS**

**Projeto 1**

Antonio José Gadelha de Albuquerque Neto(ajgan)

Rielson Leandro da Silva de Lima(rlsl)

**Recife, 2018.2**

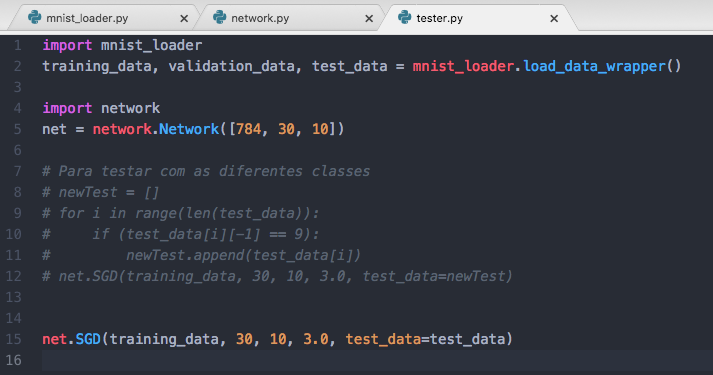
Os resultados foram obtidos usando o roteiro descrito no site da cadeira. O link fornecido nos deu uma boa visão do problema, além de terem disponibilizado o código.

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.

A rede foi treinada com os parâmetros padrões que o desenvolvedor fornece no site(epochs=30, mini\_batch\_size=10, eta=3). Percebemos que podemos mudar os parâmetros para obter resultados melhores, mas entendemos que esse não era o grande objetivo desse primeiro projeto.

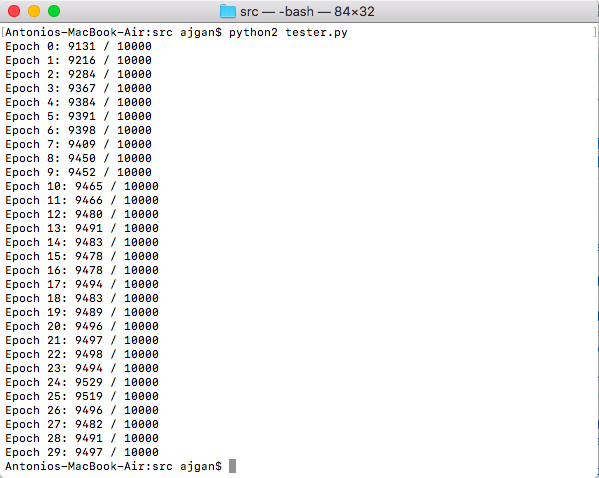
Foi feita apenas uma pequena mudança no código, para que fosse possível fazer uma filtragem nas instâncias de teste, de modo a podermos observar o resultado do aprendizado em cada uma das classes isoladas.

O código e os resultados podem ser observados no Github através do link: <https://github.com/ajgan/RN-Projeto1>

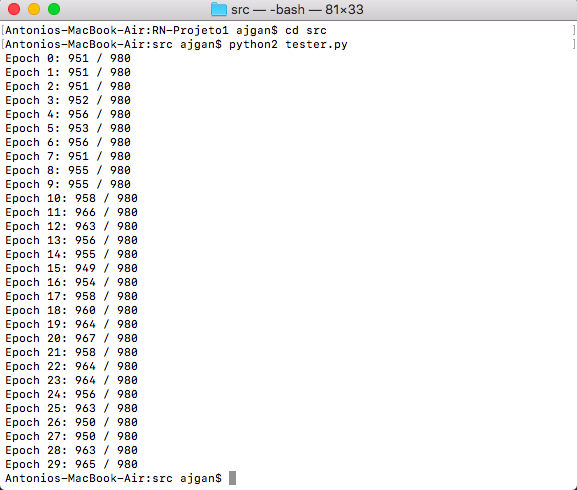


*Código-fonte. A parte comentada é a mudança que torna possível analisar as classes individualmente.*

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 classe do dígito 0. Temos prints de todas as classes, 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*



*Execução para a classe 0*

As épocas começam a contagem no 0. O total de épocas observadas foi 30 para cada caso. Por isso a última época é a 29. Vale lembrar que como a inicialização dos pesos é randômica, os resultados não seriam iguais em outras execuções.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test**  **Data** | **Resultado da Última Época(29)** | **Melhor**  **Época** | **Resultado da Melhor Época** |
| Tudo | 94,97% | 24 | 95,29% |
| Classe 0 | **98,47%** | 20 | 98,67% |
| Classe 1 | 98,24% | 20/25 | **98,94%** |
| Classe 2 | 94,28% | 14/16 | 95,45% |
| Classe 3 | 94,26% | 22 | 96,24% |
| Classe 4 | 95,72% | 16 | 96,64% |
| Classe 5 | 92,49% | 28 | 95,29% |
| Classe 6 | 97,08% | 6 | 97,70% |
| Classe 7 | 94,55% | 13/24 | 96,11% |
| Classe 8 | 95,17% | 16/19 | 95,38% |
| Classe 9 | 91,87% | 17 | 93,96% |

Os resultados observados foram bem satisfatórios. Claro que entendemos que resultados bem melhores podem ser atingidos, mas por se tratar se de uma rede simples e também por não termos nos preocupado em ajustar os parâmetros, os resultados acabam ficando dentro do esperado.