**USP/ICMC**
**SCC0270 - Redes Neurais e Aprendizado Profundo**
**Prof. Bruno S. Faiçal**
**1º semestre de 2020**

# Trabalho Avaliativo

**Grupo:**

Alexandre Norcia Medeiros - nUSP: 10295583
Daniel Penna Chaves Bertazzo - nUSP: 10349561
Vinícius Torres Dutra Maia da Costa - nUSP: 10262781

## Sumário

## 1. Introdução

O trabalho consiste na utilização das técnicas de aprendizado de máquina, especificamente redes neurais e máquinas de vetores de suporte, para classificação de dois conjuntos de dados especificados. Por meio de um método de validação e do entendimento dos respectivos conjuntos de dados, o objetivo é especificar as técnica e suas configurações (parâmetros) e comparar os seus desempenhos, de forma a obter a melhor classificação possível.

## 2. Tecnologias e Bibliotecas

Foi usado a linguagem de programação *Python* (versão 3.7) para o desenvolvimento da parte prática do trabalho. Além disso, foram utilizadas as bibliotecas *Numpy* e *Scikit-learn* do *Python*. O pacote do *Numpy* contém implementações de estruturas de dados eficientes, como vetores e matrizes, além de funções convenientes para aplicar e manipular essas estruturas. Já a biblioteca *Scikit-learn*, também conhecida como *sklearn*, contém diversos módulos com implementações de técnicas de aprendizado de máquina e métodos de validação, ainda sendo uma biblioteca de código aberto.

A linguagem *Python* é interpretada, por isso não existe um executável compilado. A execução é feita por meio de um terminal que já tenha instalado uma versão compatível do *Python* e as bibliotecas requeridas. Durante a execução, o modelo é criado, com os parâmetros já especificados, treinado e testado, utilizando o método de validação *cross-validation*, e ao final, a acurácia e outras medidas de avaliação são salvas em um arquivo texto.

## 3. Metodologia e Desenvolvimento

Os conjuntos de dados foram obtidos por meio da biblioteca Sklearn, que já os possui no módulo "*datasets*". O conjunto *Iris* é obtido pela função "`load_iris`", que retorna o conjunto numa estrutura própria da biblioteca semelhante a um dicionário do Python. O conjunto MNIST é obtido de forma similar, através da função "`load_digits`".

Para cada conjunto, foram desenvolvidos dois modelos diferentes de aprendizado de máquina com arquiteturas específicas, cujo objetivo é classificar as instâncias dos *datasets* ao analisar suas características. Um modelo é uma Rede Neural do tipo *Multilayer Perceptron* e outro é uma *Support Vector Machine*. Essas técnicas foram avaliadas de acordo com seu desempenho, com intuito de compará-las e definir qual técnica realiza a melhor separação/classificação dos dados.

Para a avaliação dos modelos, foi utilizada a técnica de *cross-validation*. Nesta técnica, o conjunto de dados é dividido em um número "n" arbitrário de subconjuntos (para o projeto foi definido "n = 10"). Os subconjuntos são estratificados, ou seja, são definidos de

forma que tenham instâncias aleatórias do conjunto original e as distribuições de classes sejam proporcionais às do conjunto original. Dessa forma, é possível selecionar os subconjuntos para o treinamento do modelo e os subconjuntos para a validação do modelo. Com o número de subconjuntos igual a 10, são criados 10 modelos, cada um utilizando 1 dos subconjuntos para a validação e os outros 9 para treinamento.

Ao final do treinamento e avaliação dos modelos, obtém-se 10 medidas de erro empírico e erro estimado para as duas arquiteturas definidas. Com isso será calculada a média desses erros e o desvio padrão (intervalo de confiança).

A definição das arquiteturas ocorreu por meio de testes durante as execuções, com seus parâmetros sendo ajustados conforme os erros obtidos após as avaliações. O *learning rate* baixo de 0.01 se mostrou eficiente para resultados constantes, diminuindo a importância de uma boa inicialização do modelo.

### 3.1 Arquitetura para o conjunto Iris

Para o *dataset* Iris, a arquitetura das redes neurais foi definida com duas camadas escondidas, a primeira contendo 5 neurônios e a segunda contendo 3, além das camadas já definidas de entrada (número de atributos) e saída (número de classes). A função de ativação é uma função sigmóide (logística) e o *learning rate* é de 0.01. Foi definido como 2000 o número máximo de iterações para convergência. Já a *Support Vector Machine* possui um *kernel* radial com uma tolerância de $10^{-4}$, que define o momento de parar o treinamento.

### 3.2 Arquitetura para o conjunto MNIST

Para o *dataset* MNIST, a arquitetura de MLP utilizada possui 2 camadas escondidas com 400 neurônios cada, função de ativação sigmóide (logística) e *learning rate* constante com valor 0.01. Foi definido como 1000 o número máximo de iterações para convergência. Já a SVM possui um *kernel* polinomial de grau 3 e uma tolerância de $10^{-5}$, que define o momento de parar o treinamento.

## 4. Resultados e Conclusões

Os resultados foram obtidos por meio da aplicação do método de validação *cross-validation* com 10 *folds* (subconjuntos), gerando 10 valores absolutos de acurácia. Dessa forma, obteve-se a média da acurácia da etapa de treino e da etapa de teste e o intervalo de confiança de 95%, além da matriz de confusão, para as duas arquiteturas definidas.

### 4.1 Resultados do conjunto Iris

Os resultados obtidos no conjunto de dados Iris foram:

| | **MLP - 2 camadas escondidas (5, 3) e eta = 0.01** | **SVM - kernel radial** |
|---|---|---|
| **Acurácia empírica (etapa de treinamento)** | $0.985 \pm 0.004$ | $0.974 \pm 0.004$ |
| **Acurácia estimada (etapa de validação)** | $0.973 \pm 0.033$ | $0.973 \pm 0.020$ |

Observando estes resultados, é possível concluir que os dois classificadores conseguem realizar a separação dos dados de maneira eficiente. Não há *underfitting*, já que as acurácias empíricas são altas o suficiente, o que implica em erros empíricos bem abaixo de 50%. Não há, também, *overfitting*, pois os valores das acurácias estimadas são altos, implicando em erros estimados muito menores que 50%.

Por fim, de acordo com os resultados, ambos os classificadores indicam uma taxa de generalização adequada do padrão aprendido, uma vez que os valores de acurácia empírica e estimada são semelhantes, em cada um dos modelos. Pode-se demonstrar esse balanceamento no conhecimento fazendo a diferença entre os erros:

Para a MLP: $Remp - Rest = 0.015 - 0.027 = -0.012 \approx 0$

Para a SVM: $Remp - Rest = 0.026 - 0.027 = -0.001 \approx 0$

Além disso, nota-se que o valor do desvio padrão dos resultados é baixo, o que indica uma consistência na etapa de treinamento dos classificadores.

As matrizes de confusão de cada modelo foram representadas em *heatmaps*:

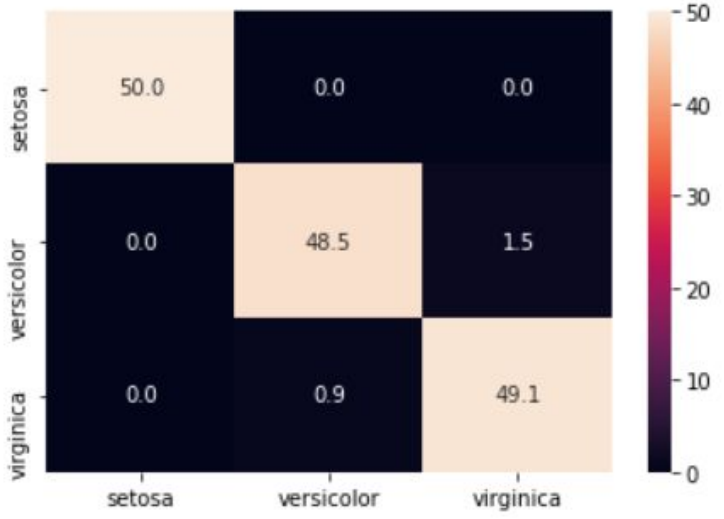


Imagem 1: Matriz de confusão média das MLPs com o conjunto Iris

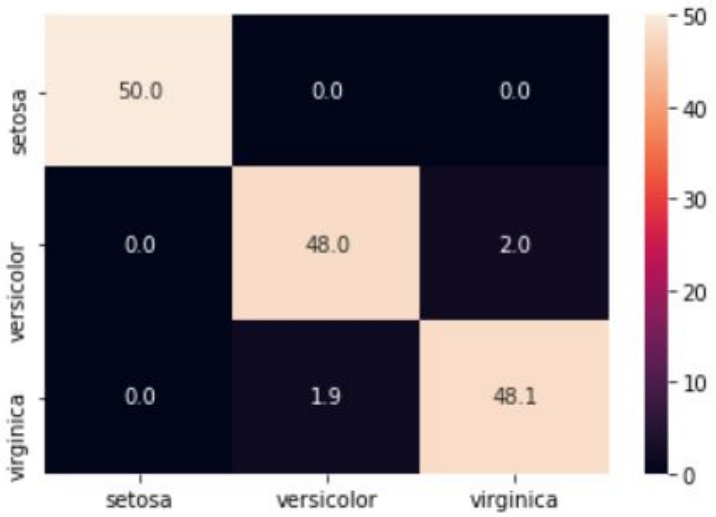


Imagem 2: Matriz de confusão média das SVMs com o conjunto Iris

### 4.2 Resultados do conjunto MNIST

Os resultados obtidos no conjunto de dados MNIST foram:

| | **MLP - 2 camadas escondidas (400, 400) e eta = 0.01** | **SVM - kernel polinomial com grau 3** |
|---|---|---|
| **Acurácia empírica (etapa de treinamento)** | $0.999 \pm 0.001$ | $0.999 \pm 0.000$ |
| **Acurácia estimada (etapa de validação)** | $0.953 \pm 0.016$ | $0.978 \pm 0.013$ |

Pode-se realizar uma análise similar à do primeiro conjunto de dados, em que ambos classificadores alcançaram um bom desempenho e uma taxa de generalização satisfatória do padrão aprendido, indicado pela taxa elevada de acurácia nas duas etapas.

Nota-se que não ocorreu *underfitting* nem *overfitting*, além da taxa de generalização ser adequada, demonstrando da mesma forma, pela diferença dos erros empíricos e estimados:

Para a MLP: $Remp - Rest = 0.001 - 0.047 = -0.046 \approx 0$

Para a SVM: $Remp - Rest = 0.001 - 0.022 = -0.021 \approx 0$

Além disso, nota-se que o valor do desvio padrão dos resultados é baixo, o que indica uma consistência na etapa de treinamento dos classificadores.

Por fim, comparou-se a SVM com kernel polinomial de grau 3, a qual é o melhor classificador construído neste trabalho para o conjunto MNIST, com as propostas citadas no site oficial. Com base nas taxas de erro, o modelo superou 18 propostas.

A seguir estão representadas as matrizes de confusão de cada arquitetura:

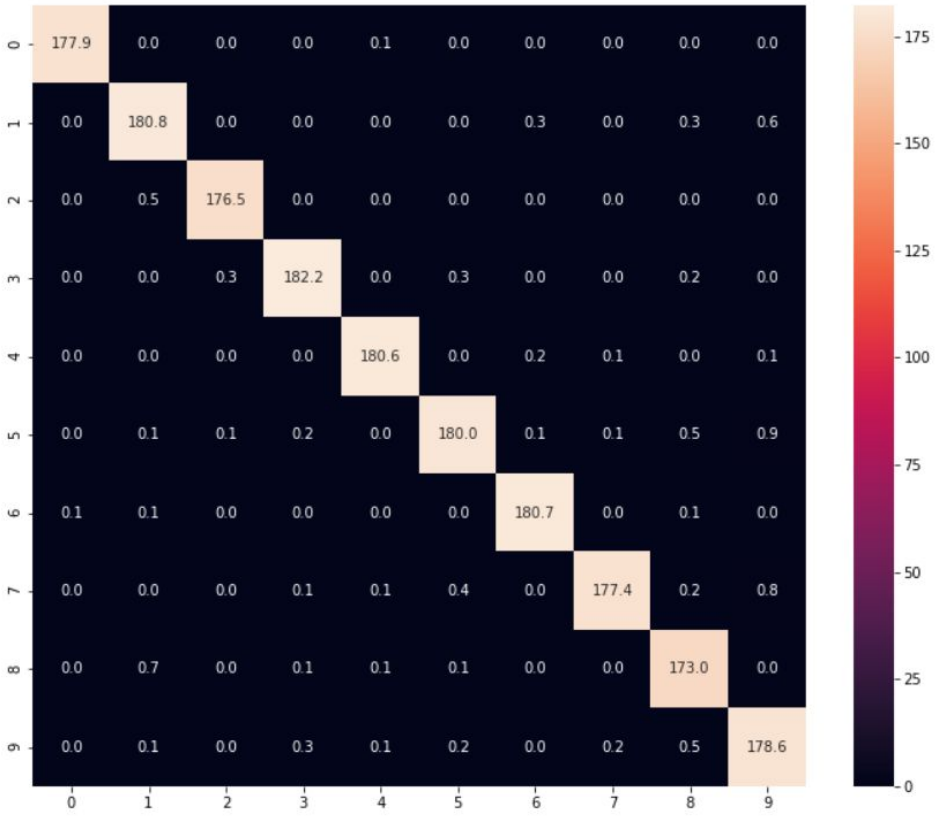


Imagem 3: Matriz de confusão média das MLPs com o conjunto MNIST

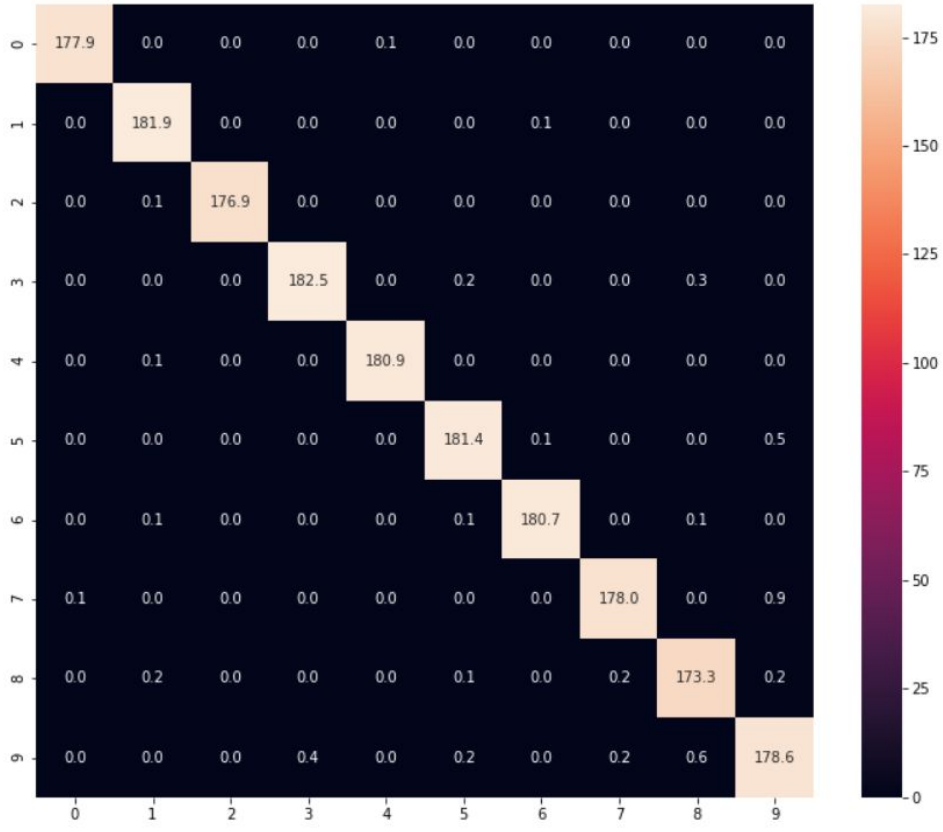


Imagem 4: Matriz de confusão média das SVMs com o conjunto MNIST