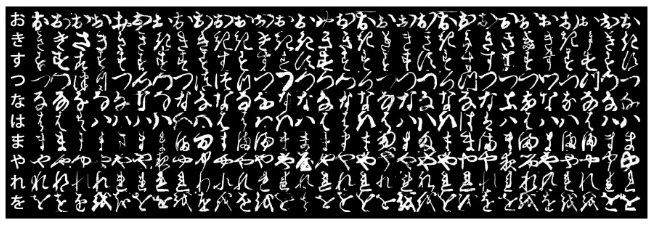
**PSI3471 - Fundamentos de Sistemas Eletrônicos Inteligentes**

Lição de Casa 6 (Aulas 11 e 12)

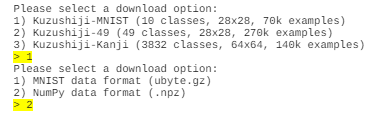
| **Nome Completo** | **Número USP** |
| --- | --- |
| Gustavo Henrique da Silva Amaral | 12551686 |
| Thiago da Rocha Calomino Gonçalves | 12554647 |

**Lição de casa #6 (Parte única)**

O site <https://github.com/roiscodh/kmnist> traz Kuzushiji-MNIST, um conjunto de imagens de caracteres japoneses manuscritos distorcidos, que pode substituir diretamente MNIST, pois consiste de 60.000 + 10.000 imagens 28×28.



Para baixar esse conjunto, basta copiar e executar a célula Python fornecida em: <https://github.com/rois-codh/kmnist/blob/master/download_data.py>. Quando executar a célula, faça as seleções como abaixo para baixar os arquivos no seu diretório default:



Depois, para carregar o BD, basta executar:

AX = np.load("kmnist-train-imgs.npz")['arr\_0']

AY = np.load("kmnist-train-labels.npz")['arr\_0']

QX = np.load("kmnist-test-imgs.npz")['arr\_0']

QY = np.load("kmnist-test-labels.npz")['arr\_0']

Classifique este banco de dados usando redes neurais densa e convolucional, **usando a função de perda sparse\_categorical\_crossentropy** em vez de *categorical\_crossentropy* (não vai precisar mais calcular *one-hot-encoding* com função *to\_categorical*, veja anexo A.6.b2 da apostila densakerasead).

Pode usar como modelos os programas:

* Densa: Programa 4 da apostila densakeras-ead (mlp1.py)
* Convolucional: Programa 1 desta apostila (cnn1.py)

Ajuste os programas para obter as menores taxas de erro possíveis. Compare as taxas de erros das redes densa e convolucional. Quem obtiver taxa de erro baixa recebe nota mais alta.

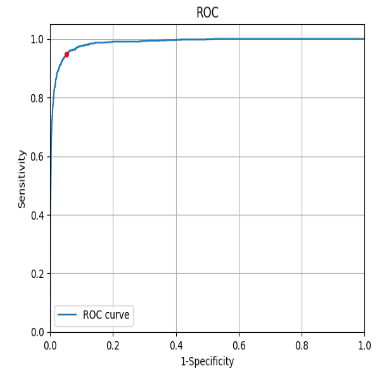
Solução privada:

* Densa: <https://colab.research.google.com/drive/1MEBijyqaCw8eYUvi3QA_ii3N0xh722D-#scrollTo=a4aVP3czWi1v>
* Convolucional: <https://colab.research.google.com/drive/13wQwGNgh3Y-DXLplVPdA2CckwnITfGen#scrollTo=a4aVP3czWi1v>
* Densa erro de teste: 9.02 %
* Convolucional erro de teste: 4.35 %

**Exercício:** vamos supor que as imagens de pulôver do fashion-mnist representam exames de imagem de pacientes “doentes” e as demais representam exames de pacientes “sadios”. Modifique o programa que classifica fasion-mnist para que identifique se um paciente está “doente” ou “sadio”, classificando o seu exame em “pulôver” ou “não-pulôver”. Treine o seu modelo com as imagens de treino do fashion-mnist e faça a predição usando as imagens de teste. Adote limiar=0.5 e calcule sensibilidade e especificidade. Plote curva ROC e calcule AUC. Calcule acuracidade no ponto de EER (equal error rate).

* <https://en.wikipedia.org/wiki/Sensitivity_and_specificity>
* <https://en.wikipedia.org/wiki/Receiver_operating_characteristic>
* <https://www.w3schools.com/python/python_ml_auc_roc.asp>
* <https://scikit-learn.org/stable/auto_examples/model_selection/plot_roc.html>

A curva ROC obtida deve ser semelhante a:



Solução privada: <https://colab.research.google.com/drive/1wtjaLvzKBfGy_jK5YBs-EOLrF_tnMo_i#scrollTo=YXec8kUmpTl>

**Explicação completa:**

-

**Código da lição:**

| // Lição de casa 6 (Parte Única)  //  // Gustavo Henrique da Silva Amaral - 12551686  // Thiago da Rocha Calomino Gonçalves - 12554647  // . . . |
| --- |