



Universidade do Minho
Departamento de Informática
Mestrado (Integrado) em Engenharia Biomédica

Mestrado em Engenharia Informática
Aprendizagem Profunda
1º Ano, 2º Semestre
Ano letivo 2024/2025

Enunciado Prático
10 de abril de 2025

Tema	<i>Classificação de imagens RGB com CNNs usando CIFAR-10</i>
Enunciado	Pretende-se, com esta ficha, que seja realizado um conjunto de tarefas que permitam consolidar o conhecimento adquirido sobre Redes Neurais Convolucionais.
Tarefas	<p>Utilizando o <i>framework</i> que preferirem (e usando, se possível, o <i>cuda</i>) devem, numa primeira fase executar as seguintes tarefas (<u>6.CNN_multiclass_CIFAR10.ipynb</u>):</p> <p>T1 – Fazer <i>download</i> das imagens do <i>dataset</i> CIFAR-10, que consiste em 60000 imagens a cores com 32x32 pixels (50000 imagens para treino, 10000 imagens para teste):</p> <p>Executar</p> <pre>wget http://pjreddie.com/media/files/cifar.tgz</pre> <pre>tar xzf cifar.tgz</pre> <p>T2 – Aplicar as instalações de dependências necessárias (<i>nvcc</i>, <i>torch</i>, <i>liveplot</i>, ...).</p> <p>T3 – Definir um <i>batch_size</i> de 128.</p> <p>T4 – Na preparação dos dados:</p> <p>T4.1 – Contruir uma lista com as classes do <i>dataset</i>.</p> <p>T4.2 – Aplicar um pré-processamento às imagens, normalizando-as.</p> <p>T4.3 – Mudar o canal de cores para CHW em vez de WHC.</p> <p>T4.4 – Ler o <i>dataset</i> e preparar os <i>data loaders</i> com <i>holdout</i>.</p> <p>T5 – Para visualizar os dados em conformidade, colocar primeiramente o <i>label</i> por extenso. Depois, imprimir as métricas relevantes do <i>dataset</i> e apresentar um <i>batch</i> dos casos de treino para verificar o que se vai trabalhar.</p> <p>T6 – Verificar se o <i>dataset</i> está balanceado, incluindo os casos de validação.</p>

T7 – Definir os vários modelos para aplicar:

T7.1 – Modelo *ResNet*: modelo com vários blocos residuais e várias camadas:

- *Convolutional Layer*
- *Batch Normalization Layer*
- *Sequential Composed Layer*
 - *Residual Block*
 - *Convolutional Layer*
 - *Batch Normalization Layer*
 - *Convolutional Layer*
 - *Batch Normalization Layer*
 - *Sequential Layer*
 - *Residual Block*
 - *Convolutional Layer*
 - *Batch Normalization Layer*
 - *Convolutional Layer*
 - *Batch Normalization Layer*
 - *Sequential Layer*
- *Sequential Composed Layer*
- *Sequential Composed Layer*
- *Sequential Composed Layer*
- *Linear Layer*

Fazer *download* do modelo da BB ([CNNModel_cifar_Resnet.pth](#)).

T7.2 – Modelo *CNN 1*:

- *Sequential Composed Layer*
 - *Convolutional Layer*
 - *ReLU activation*
 - *Max Pooling*
- *Sequential Composed Layer*
- *Linear Layer*
- *ReLU activation*
- *Linear Layer*
- *Softmax activation*

Fazer *download* do modelo da BB ([CNNModel_cifar_1.pth](#)).

T7.3 – Modelo *CNN 2*:

- *Sequential Composed Layer*
 - *Convolutional Layer*
 - *ReLU activation*
 - *Max Pooling*
- *Sequential Composed Layer*
- *Linear Layer*

Fazer *download* do modelo da BB ([CNNModel_cifar_2.pth](#)).

T7.4 – Modelo *CNN 3*:

- *Sequential Composed Layer*
 - *Convolutional Layer*
 - *Batch Normalization Layer*
 - *ReLU activation*
 - *Max Pooling*
- *Sequential Composed Layer*

- *Linear Layer*
- *Dropout*
- *Linear Layer*
- *Linear Layer*

Fazer *download* do modelo da BB ([CNNModel_cifar_3.pth](#)).

T7.5 – Modelo CNN 4:

- *Sequential Composed Layer*
 - *Convolutional Layer*
 - *Batch Normalization Layer*
 - *ReLU activation*
 - *Max Pooling*
 - *Dropout*
- *Linear Layer*
- *Linear Layer*

Fazer *download* do modelo da BB ([CNNModel_cifar_4.pth](#))

T8 – Treinar todos os modelos em separado, registrando os valores de *accuracy* e *loss* para os casos de treino e para os casos de validação.

T8.1 – Modelo *ResNet*: usar 30 *epochs*, uma *learning rate* de 0.001, usar as funções *Cross Entropy Loss* e a *SGD*.

T8.2 – Modelo CNN 1: usar 15 *epochs*, uma *learning rate* de 0.001, usar as funções *Cross Entropy Loss* e a *SGD*.

T8.3 – Modelo CNN 2: usar 15 *epochs*, uma *learning rate* de 0.001, usar as funções *Cross Entropy Loss* e a *SGD*.

T8.4 – Modelo CNN 3: usar 15 *epochs*, uma *learning rate* de 0.001, usar as funções *Cross Entropy Loss* e a *SGD*.

T8.5 – Modelo CNN 4: usar 75 *epochs*, uma *learning rate* de 0.001, usar as funções *Cross Entropy Loss* e a *SGD*.

T9 – Avaliar os modelos em separado, apresentando as previsões e a matriz de confusão.

T10 – Usar os modelos em separado, tentando prever o caso dado.

T11 – Submeter um notebook em que se indica, para cada tarefa e para cada modelo, o esquema da rede criada, os hiper-parâmetros utilizados, os valores de *accuracy* e de *loss* obtidos, e os gráficos de aprendizagem.