

MAC05921 – *Deep Learning*

DCC / IME-USP — 2025

Tarefa 2

Data de entrega: 07/09/2025

1. Esta tarefa supõe que você já está minimamente familiarizado(a) com o **PyTorch**.
2. **Objetivo:** implementar e praticar o treinamento de Redes Neurais convencionais (NN) e Redes Neurais Convolucionais (CNN) – usando **PyTorch**, em tarefas de classificação de imagens
3. **Exploração mínima a ser feita**
 - (a) Implementar e treinar NN e CNN (não muito grandes), preferivelmente com número de parâmetros treináveis similares.
 - (b) Sugestão de dataset: MNIST (sim, é um dataset simples, mas é adequado para uma prática inicial com NN \times CNN)¹
 - (c) Inicialmente considere apenas dados de treinamento e número de épocas de treinamento fixo. Use apenas parte do conjunto de treinamento (por exemplo 300 exemplares de cada dígito), para não “gastar” muito tempo na exploração. Veja se há convergência da loss, com que velocidade a loss converge, para que valor converge, se chega a se estabilizar, etc. Repita o experimento variando o número de exemplares por classe, se necessário.
 - (d) Em seguida, considere um conjunto de validação, e utilize-o para monitorar overfitting, para fazer early stopping. Compare os desempenhos no treinamento e na validação.
 - (e) Observe se há e anote as diferenças significativas entre o comportamento e desempenho de NNs e CNNs nos experimentos que realizou
4. **Sugestão de explorações adicionais**
 - (a) (NN e CNN) Considere imagens ligeiramente modificadas: por exemplo, dígitos de tamanhos variados, posicionamentos distintos dentro do domínio 28×28 , presença de ruídos do tipo sal e pimenta².
 - (b) (NN e CNN) Considere classes bem desbalanceadas e observe como muda ou não o comportamento e desempenho dos modelos. Pode-se também considerar um número menor de classes.

¹Caso você já tenha experiência sólida em treinar NNs e CNNs, poderá ir diretamente para as **Explorações adicionais**.

²Se você implementar essas modificações, fique à vontade para compartilhar as imagens modificadas ou o código usado para gerar as modificações.

- (c) (NN e CNN) Considere visualizações como curvas de loss, matrizes de confusão, t-SNE plot
- (d) (CNN) Se você já está bem familiarizado(a) com CNNs, e se tem acesso sem muita restrição a GPUs, fique à vontade para explorar datasets de classificação de imagens mais complexos
- (e) (CNN) Considere a aplicação de GradCAM para visualizar quais partes da imagem estão levando à ativação maior de determinada saída
- (f) Exploração de hiperparâmetros
 - número de parâmetros treináveis na rede
 - número de camadas, número de nós/filtros por camada, tamanho do kernel, etc
 - tipos de ativação, normalização, regularização, inicialização de pesos
 - batch size, learning rate, otimizador, etc

O que entregar

ENTREGAR código fonte desenvolvido e um relatório simples em PDF.

O **relatório** deve ter no máximo 2 páginas em fonte normal. Deve descrever de forma sucinta, organizada e compreensível o que foi realizado. A descrição deve contemplar a implementação (se foi implementação própria – consultando o manual do PyTorch, etc, ou se foi baseado em códigos prontos disponíveis na web – neste caso devem ser informadas as fontes e quais adaptações foram feitas, se for o caso), as explorações realizadas, e o que aconteceu/foi observado nas explorações.

Para treinar as redes e fazer previsões, pode-se usar GPUs no Google Colab ou no Kaggle, por exemplo. Devido às retrições de tempo de uso, é importante planejar bem antes de sair apertando botões.

Dúvidas

Postem as dúvidas no Fórum de discussões no e-disciplinas. Trechos de código podem ser postados seja para pedir esclarecimentos sobre certas construções de código, seja para explicar ou compartilhar trechos específicos.