

MAC05921 – *Deep Learning*

DCC / IME-USP — 2024

EP 1

Data de entrega: 08/09/2024

O objetivo desta tarefa é o exercício prático de conceitos, técnicas e métodos vistos ou relacionados ao conteúdo coberto nas aulas. Especificamente, o foco será a familiarização prática com Redes Neurais convencionais (NN) e Redes Neurais Convolucionais (CNN).

O que fazer

Espera-se que vocês treinem e avaliem arquiteturas de redes *fully connected* e CNNs sobre um mesmo problema de classificação de imagens.

Se você nunca treinou uma rede neural ou uma CNN, pode usar alguma implementação disponível na web e expandir/modificar a partir dela. Vocês podem inclusive criar um notebook v_0 em pares ou pequenos grupos, e em seguida cada um faz o seu caminho.

Datasets

Esses são datasets bem conhecidos e relativamente “fáceis”. Fiquem à vontade para usar outro dataset se assim desejar. Porém, tomem cuidado com o tamanho das imagens, pois o treinamento pode ficar demorado.

- MNIST (<http://yann.lecun.com/exdb/mnist/>)
- Fashion-MNIST <https://github.com/zalandoresearch/fashion-mnist>
- Kuzushiji-MNIST <https://github.com/rois-codh/kmnist>
- MNIST-MIX <https://github.com/jwwthu/MNIST-MIX>
- CIFAR-10 <https://www.cs.toronto.edu/~kriz/cifar.html>

Uma dica: familiarize-se com os dados; veja os exemplos, gere estatísticas, entenda as características do conjunto de dados, antes de definir arquiteturas ou fazer treinamento.

Obrigatório: Exploração mínima esperada

Como critério mínimo para a aceitação da entrega, o aluno deve responder a seguinte questão:

- Como a performance de uma rede neural totalmente conectada se compara com a de uma CNN ao classificar imagens de um dataset específico?

As implementações devem possuir uma quantidade similar de parâmetros treináveis.

Existem comportamentos que podem ser observados, dependendo das explorações feitas, tais como *overfitting*, velocidade de convergência (quantas épocas até a estabilização da loss). Isso pode ser visto plotando, por exemplo, a curva de aprendizado (época x loss).

Mesmo que você não consiga responder completamente a pergunta, é importante que você entregue o seu trabalho dentro do prazo estabelecido. O prazo não será estendido, portanto, por favor, faça o melhor que puder dentro do tempo disponível. Entregar o trabalho parcial ainda é valioso e pode ser avaliado com base no progresso que você fez. Se você encontrar dificuldades, informe-as claramente e descreva o que foi feito até o momento.

Exploração adicional

Durante a realização desta tarefa, é esperado que o aluno desenvolva uma postura crítica e investigativa, formulando suas próprias perguntas e levantando dúvidas sobre o problema proposto.

Abaixo seguem algumas perguntas para guiar a exploração. No entanto, é importante destacar que o aluno não precisa se limitar estritamente a essas questões. A liberdade de explorar diferentes abordagens e de seguir suas próprias curiosidades é não apenas permitida, mas encorajada.

- Qual é a distribuição das classes no dataset que você escolheu? Existem classes desbalanceadas?
- Quais são as principais diferenças entre uma rede neural totalmente conectada (fully connected) e uma rede neural convolucional (CNN) em termos de arquitetura?
- Como as operações de convolução em uma CNN ajudam na extração de características de uma imagem?
- Qual é o papel das camadas de pooling em uma CNN? Seria possível treinar uma CNN sem elas? O que aconteceria?
- Como você pode visualizar e interpretar os filtros (kernels) aprendidos por uma CNN?
- O que essas visualizações dizem sobre o tipo de características que a CNN aprende nas primeiras camadas versus nas últimas?
- É possível interpretar os pesos de uma rede totalmente conectada da mesma forma? Por que ou por que não?

Além disso, há diversos hiperparâmetros e configurações do problema que podem ser explorados. Escolha alguma exploração que ache interessante. Mais do que tentar ser exaustivo, pode ser mais interessante focar em algum aspecto específico na exploração.

- 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
- número de classes, número de exemplos de treinamento em cada classe (sim, podem ser explorados casos com desbalanceamento)
- “aumentação” de dados, redução do tamanho da imagem, etc

O que entregar

Deve ser entregue o código fonte utilizado na exploração e um relatório simples com as seguintes informações.

Caso o código entregue esteja em um notebook, o relatório pode estar no próprio notebook ou podem estar em um PDF à parte.

No início do relatório coloque sua identificação (nome) e um resumo do trabalho realizado. Procure escrever de forma clara e objetiva.

Também deve ser informado:

- todas as fontes utilizadas: notebooks de terceiros, páginas consultadas e que efetivamente serviram como referência, colegas, etc.
- Dificuldades enfrentadas.
- Perguntas formuladas para as quais ainda não chegou a uma resposta.
- Qualquer coisa positiva associada à execução desta tarefa.

Plataforma a ser usada na prática

Para fornecermos um feedback mais eficiente, sugerimos a utilização do PyTorch¹. No entanto, fique à vontade para usar outra biblioteca que preferir. Nestes casos, o feedback pode levar um pouco mais de tempo.

Para treinar a rede e fazer predições, é relevante termos acesso a GPUs. Recomendamos o uso do Google Colab.

¹Até onde eu entendo, o Keras/Tensorflow é mais encapsulado e para fazer coisas padrões é muito fácil. Já para fazer customizações, dizem que o Pytorch é muito mais amigável. Então entendo que se você quer ir além de ser um “apertador de botão”, o PyTorch seja melhor. Mas, nada contra descer fundo no TensorFlow.