Deep Learning (IST, 2025-26)

Homework 2

Margarida Campos, Guilherme Coimbra, Fábio Faria, Chrysoula Zerva

Deadline: Wednesday, January 7, 2026.

This homework should be submitted in the format of a short 2-column article, of max length 6 pages, following the adapted ARR template, available here: `https://www.overleaf.com/read/hndryfsnzqmz#00ce46`

Please submit **a single zip file** containing the code files and the report in Fenix under your group's name.

Question 1 (15 points)

Image classification with CNNs. In this exercise, you will implement a convolutional neural network to perform classification using the BloodMNIST[1] dataset. It is based on a dataset of individual normal cells, captured from individuals without infection, hematologic or oncologic disease and free of any pharmacologic treatment at the moment of blood collection. It contains a total of $17,092$ images and is organized into 8 classes. The source dataset has been splitted with a ratio of 7:1:2 into training, validation and test set. The source images with resolution $3 \times 360 \times 363$ pixels are center-cropped into $3 \times 200 \times 200$, and then resized into $3 \times 28 \times 28$.

As previously done in Homework 1, you will need to download the file `bloodmnist.npz` on Zenodo repository.[2] You can also do this by running the following command in the homework directory:

```
pip install medmnist
```

Next, import the following libraries:

```
from torch.utils.data import DataLoader
from torchvision import transforms
from medmnist import BloodMNIST, INFO
```

Finally, the Python skeleton code is provided (`hw2-q1.py`).

For this exercise, we recommend you use a deep learning framework with automatic differentiation.

Hint: We recommend using PyTorch and the functions of the nn.Module class

1. (10 points) Implement a simple convolutional network with the following structure:

    - A convolution layer with 32 output channels, a kernel of size 3×3, a stride of 1, and a padding of 1.

[1] BloodMNIST - `https://medmnist.com/`

[2] Zenodo repository - `https://zenodo.org/records/10519652`

- A ReLU activation function.
- A convolution layer with 64 output channels, a kernel of size 3×3, stride of 1, and a padding of 1.
- A ReLU activation function.
- A convolution layer with 128 output channels, a kernel of size 3×3, stride of 1, and a padding of 1.
- A ReLU activation function.
- A linear layer with 256 output features (to determine the number of input features use the number of channels, width and height of the output of the **third** block. Hint: The number of input features = $number\_of\_output\_channels \times output\_width \times output\_height$).
- A ReLU activation function.
- A linear layer with the number of classes followed by an output WITH and WITHOUT a Softmax layer.

Train your model for 200 epochs using `optim.Adam()`, the `nn.CrossEntropyLoss()` function, batch size of 64, and learning rate of 0.001 **on the training set and evaluate it on the validation and test sets.** Compare and briefly discuss the results achieved with and without Softmax layer, i.e., $Logits \times Softmax$**, reporting the test accuracy of the model with the best performance on the validation set.** Furthermore, you will plot **two curves**: the training loss **and** the validation accuracy, both as a function of the epoch number.

2. (5 points) Now, you will add 3 layers `nn.MaxPool2d(2)`, each after a ReLU activation function of the convolutional block. **Thus, execute the same experimental protocol adopted previously in (1), however with the new models** $Logit + Maxpool$ **and** $Softmax + Maxpool$**.** Now you will discuss the impact of the Maxpooling layers in terms of effectiveness (accuracy) and efficiency (time).
Hint: Use `import time` *to measure elapsed time in Python.*

Question 2 (85 points)

RNA Binding Protein (RBPs) Interaction Prediction. Cells are the primary unit of life for all organisms. A cell's composition includes proteins and nucleic acids (DNA and RNA), among other molecules. **DNA** encodes the genetic information responsible for the development and functioning of an organism. It has a double helix structure made of two linked strands, each composed of a linear sequence of 4 chemical bases: Adenine (**A**), Thymine (**T**), Guanine (**G**), and Cytosine (**C**). The two strands are connected by chemical bonds between the bases: A bonds with T, and C bonds with G. The sequence of these bases along the DNA backbone encodes biological information [Alberts et al., 2002]. **RNA** molecules are single-stranded and made of a linear sequence of bases: **A**, **C**, **G**, and Uracil (**U**). Because DNA cannot be decoded directly into proteins, it is first transcribed into messenger RNA (**mRNA**) [Crick, 1970]. The life cycle of eukaryotic mRNA involves several stages: transcription, processing, export, translation, and decay. In each stage, mRNA interacts with a specific set of **RNA Binding Proteins (RBPs)**. These RBPs govern the activity and stability of mRNA. An RBP acts like a key seeking a specific lock; it scans RNA molecules for specific patterns (**motifs**) or structural shapes that match its binding domain [Gerstberger et al., 2014].

The interaction between an RBP and an RNA sequence is determined by the **binding affinity** — a continuous measure of how strongly the protein binds to the RNA.

Understanding which RNA sequences a specific RBP binds to is crucial. For example, the splicing factor **RBFOX1** binds with high affinity to the motif `UGCAUG`. Mutations in this sequence can disrupt gene splicing and are linked to neurodevelopmental disorders [Ray et al., 2013, Conboy, 2017].

We treat this biological challenge as a regression problem. The task is to learn a function f:

$$f : \Sigma^L \to \mathbb{R} \tag{1}$$

Where $\Sigma = \{A, C, G, U\}$ is the RNA alphabet, L is the sequence length, i.e. the number of nucleotides (padded to a fixed size), and the output is a continuous scalar value representing the binding intensity (affinity).

We will use data derived from the **RNAcompete** protocol, a massive collection of *in vitro* experiments [Ray et al., 2013]. This dataset consists of approximately 241,000 synthetic RNA sequences, each ranging from 38 to 41 nucleotides in length. These sequences were computationally designed using De Bruijn sequences to ensure that all possible 9-mer combinations are represented at least 16 times. Crucially, the data is partitioned into two non-overlapping halves: **Set A** (used for training) and **Set B** (held-out for testing), which ensures that models are evaluated on their ability to generalize to new sequences rather than memorizing overlapping k-mers.

- **Input (X):** Synthetic RNA sequences ranging from 38 to 41 nucleotides.
- **Target (Y):** Binding affinity, measured via fluorescence intensity (normalized).

The dataset and metadata are available at:

`https://drive.google.com/drive/folders/1b9FfWZqEtPEdsSDu_lWQIJiPP7Z3rBKL?usp=sharing`

We provide a `utils.py` file containing a pre-built data loader. The data is already pre-processed (One-Hot Encoded, Log-Transformed, and Z-Scored). Please consult the provided `README.md` for more details.

For this assignment, you must train your models on the protein **RBFOX1**. You can load it using the provided utility functions (e.g., `load_rnacompete_data('RBFOX1', split='train')`).

We are also providing you with the key training and evaluation components that should be adopted as indicated in the requirements below.

- **Loss Function:** You should use Mean Squared Error (MSE). *Note:* The dataset contains padded invalid entries (NaNs). You must use the `masked_mse_loss` provided in `utils.py` to avoid training on invalid data.
- **Evaluation Metric:** The primary metric for evaluation is the **Spearman Rank Correlation**. In fluorescence assays, the absolute intensity values can be noisy. We are more interested in the **ranking** capability of the model (i.e., does the model correctly predict that Sequence A binds stronger than Sequence B?) rather than the exact regression value.
- **Validation Strategy:** The data loader provides a 'train' split, a 'val' split, and a 'test' split. You must use the 'val' split to tune your hyperparameters. Do not use the 'test' set for tuning; use it only for the final reporting of results.

1. (40 points) Based on the information above, you are asked to choose and implement **two different deep neural network** architectures. You can choose among those seen in class, e.g. a CNN, RNN or transformer variant.

Hint: We propose to use functions from the PyTorch nn.Module class for the implementation of the NN variants, but you can also use the HuggingFace transformers library. Additionally, while you should not employ already trained models for the task, you are free to experiment with the addition of pre-trained embeddings, etc, if you want.

In the report, you should:

1. Justify the choice of models you decided to experiment with
2. Indicate which hyperparameters you chose to tune, and specify the range and optimisation strategy
3. Provide plots for loss on train and validation sets for
4. Compare the performance of the two different models and comment on whether it aligned with your initial expectations.

2. (30 points) Extend one of the initially chosen architectures to include attention (choose one of the models that was not already employing attention).

   Hint: You can add a self-attention block or attention-pooling to CNN or RNN architectures.

   In the report, you should:

   1. Specify which attention you are choosing to implement and the number of attention heads. Justify your choices.
   2. Explain what your expectations are on how model behaviour will change
   3. Provide plots for loss on train and validation sets with and without the use of attention.
   4. Compare the performance on the test set before and after the incorporation of attention.

3. (15 points) In this assignment you trained models to predict the binding affinity of a *single* RNA-binding protein (RBFOX1) from RNA sequence.

   Suppose you now want a model that **generalises to multiple RNA-binding proteins**, so that, given an RNA sequence and the identity (or features) of a protein, it predicts their binding affinity.

   (a) Describe how you would need to modify each of the following in order to tackle this multi-protein setting:
   - the data and labels,
   - the model architecture, and
   - the training objective and evaluation protocol.

   (b) Briefly discuss one potential **benefit** and one **challenge** of training such a model that learns to predict binding for multiple proteins simultaneously.

References

[Alberts et al., 2002] Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K., and Walter, P. (2002). *Molecular Biology of the Cell*. Garland Science, 4 edition.

[Conboy, 2017] Conboy, J. G. (2017). Developmental regulation of rna processing by rbfox proteins. *Wiley Interdisciplinary Reviews: RNA*, 8(2):e1398.

[Crick, 1970] Crick, F. (1970). Central dogma of molecular biology. *Nature*, 227(5258):561–563.

[Gerstberger et al., 2014] Gerstberger, S., Hafner, M., and Tuschl, T. (2014). A census of human rna-binding proteins. *Nature Reviews Genetics*, 15(12):829–845.

[Ray et al., 2013] Ray, D., Kazan, H., Cook, K. B., Weirauch, M. T., Najafabadi, H. S., Li, X., Gueroussov, S., Albu, M., Zheng, H., Yang, A., et al. (2013). A compendium of rna-binding motifs for decoding gene regulation. *Nature*, 499(7457):172–177.