EX 1 – DL basics

Due date: 9/12/2024, 11:59 PM

Please read the submission guidelines before you start.

Theory [50 pts]

1. [12.5 pts] Suppose you have an MLP composed of an input of size 10, followed by one hidden layer with output of size 50, and finally one output layer with 3 output neurons. All artificial neurons use the ReLU activation function. The batch size used is m.
   a. What is the shape of the input X?
   b. What about the shape of the hidden layer's weight vector W_h, and the shape of its bias vector b_h?
   c. What is the shape of the output layer's weight vector W_o, and its bias vector b_o?
   d. What is the shape of the network's output matrix Y?
   e. Write the equation that computes the network's output matrix Y as a function of X, W_h, b_h, W_o, and b_o?.
2. [12.5 pts] Consider a CNN composed of three convolutional layers, each with 3 × 3 kernels, a stride of 2, and SAME padding. The lowest layer outputs 100 feature maps, the middle one outputs 200, and the top one outputs 400. The input images are RGB images of 200 × 300 pixels. What is the total number of parameters in the CNN? Explain your answer.
3. [25 pts] In this question, we will derive the gradient for a batch normalization layer.

   The algorithm of Batch Normalization, as taken directly from the original paper by Sergey Ioffe and Christian Szegedy:

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β

Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_\mathcal{B} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma_\mathcal{B}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu_\mathcal{B})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_\mathcal{B}}{\sqrt{\sigma_\mathcal{B}^2 + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma \widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

- **BN** stands for Batch Norm.
- f - represents a layer upwards of the BN one.
- y - is the linear transformation which scales x by γ and adds β.
- $\hat{x}$ is the normalized input.
- μ is the batch mean.
- σ^2 is the batch variance

The operation steps are given by:

- $f(y)$
- $y(\hat{x}, \gamma, \beta)$
- $\hat{x}(x, \mu, \sigma^2)$

Consider a 1 dimensional BN layer with a mini batch of size m.

For every $1 \leq i \leq m$, the gradient of $\frac{\partial f}{\partial y_i}$ is given.

Using all notations given (and the chain rule), calculate:

a. $\frac{\partial f}{\partial \gamma}$

b. $\frac{\partial f}{\partial \beta}$

c. $\frac{\partial f}{\partial \hat{x}_i}$

d. $\frac{\partial f}{\partial \sigma^2}$

e. $\frac{\partial f}{\partial \mu}$

f. $\frac{\partial f}{\partial x_i}$

In final your answers, you may use $\frac{\partial f}{\partial \hat{x}_i}$ (and of course $\frac{\partial f}{\partial y_i}$) , but please take care that you are not leaving any other gradients in them, and that you end up with the clearest answer you can.

Practical [50 pts]

You will be implementing and training the Lenet5 network over the FashionMNIST dataset. There are a few variations of Lenet5 and you should make sure to use the one which is fitted for MNIST. You should use PyTorch as your framework for implementation.

The dataset is available on Canvas and can also be downloaded at:
https://github.com/zalandoresearch/fashion-mnist

You will specifically be comparing the usage of the following techniques with Lenet5:

- Dropout (at the hidden layer)
- Weight Decay (also known as $l2$ loss)
- Batch Normalization

a. A convergence graph is a graph with epochs as the x-axis, and accuracy as the y-axis. Provide a convergence graph for each of the three techniques – and for each of them plot one graph for the accuracy on the train data and one for the test. In addition, plot one graph without regularization (8 graphs in total).
b. Note: For dropout, the train accuracy must be measured without dropout.
c. Provide a table, which summarizes all 8 final accuracies.
d. Make conclusions regarding the results (both regarding individual techniques and differences between them).

Comments:

- Describe in the readme file how to train each setting, and how to test it with the saved weights.

- All graphs should be clear with a proper heading. It is highly recommended (but not mandatory) to plot the train and test graphs for each technique together in the same plot (only 4 plots in total).
- For dropout, the train accuracy must be measured without dropout.
- Refer to LeCun *et al.*, 1998 for the Lenet5 architecture applied to MNIST. You may need to make some modifications to adapt the architecture to FashionMNIST's input dimensions. Feel free to change the activation functions they used. Describe the specific architecture you implemented in your report.
- You should describe your training hyperparameters (batch size, learning rate and optimizer) and how you decided on them in your report. You should also specify how you performed your train/validation split and how you selected the model to perform evaluation with.
- Likewise, you should be explicit about hyperparameter values and how you decided on them for the regularization strategies.
- If you do not achieve at least 88% test accuracy – you're doing something wrong! (much more can be achieved).

Good Luck!