Deep learning - HW1

Submission date: 15/12/2022

1. In class, we defined ψ to be the log of the moment generating function. I.e.,

$$\psi(\lambda) \triangleq \log\left(\mathbb{E}\left[e^{\lambda L}\right]\right),$$

   where L denotes the loss function. Write the first and second derivatives of ψ.

2. Random labels – In this question, you will compare the KL divergence values of **Bayesian neural networks models** trained on MNIST with and without the label randomization process. Train the following models:
   - Without randomization
     i. Train a classifier on the full dataset.
     ii. Train a classifier on the first 200 samples of MNIST.
     iii. Train a classifier on the 200 first '3' and '8' samples.
     iv. Train a classifier on all '3' and '8' samples.
   - With randomization
     i. Use the first 200 samples of MNIST.
     ii. Generate random labels from Bernoulli distribution with a probability of ½. I.e., each sample is assigned a random zero or one label.
     iii. Train the network until convergence.

   What are the values of the KL divergence between the prior and posterior distributions for the models (both norm and average per parameter)? **Discuss the results** in your report.
3. L2 Regularization and Bayesian neural networks – On the MNIST dataset:
   - Train a logistic regression model
   - Train a logistic regression model with L2 regularization
   - Train a Bayesian neural network – train three different modifications (number of layers, hidden size).

   What are the differences between the three models (and their modifications)? Discuss your results in the report from theoretical and practical points of view.

You may use implemented Bayesian neural networks from the internet in all questions. For example:

- https://github.com/piEsposito/blitz-bayesian-deep-learning
- https://github.com/IntelLabs/bayesian-torch

Submission instructions:

Submission **must be individual** and will contain a short (two pages) pdf report containing:

- Model architecture description, training procedure (hyperparameters, optimization details, etc.).
- Convergence plot of accuracy as a function of time (epochs). The plot should depict both training and test performance (i.e. two curves, one for the train, and one for the test).
- A short summary of your attempts and conclusions.

In addition, you should also supply:

- Code (python file) able to reproduce your results - we might test it on different variants on these datasets.
- Trained networks with trained weights (.pkl file).
  [the weights tensors can be saved with *torch.save({'w1':w1, 'w2':w2 }, 'path_to_w.pkl')* and load with *torch.load('path_to_w.pkl')*]

Moodle submission:

You should submit a Zip file containing:

- Python files for each practical question:
  - Training procedure, file name: *hw1_id_train.py*
  - Evaluation procedure, file name: *hw1_id_eval.py*
- 1 pdf file with
  - Your full name and ID
  - Typed answers for the theoretical part
  - A summary of the practical part
- Pickle files (If the file is too big for the Moodle, upload it to your Google-Drive and copy the link to your pdf report)

Good Luck!