HW 2: Vision and Uncertainty

Version: April 25, 2023

You will train a simple image segmentation network and study how well calibrated its confidence is.

1. Download a small segmentation dataset and segmentation network.
2. We will do the training “by hand” without high level libraries (without keras, pytorch-lightning, huggingface, etc), see detailed instructions in code TODO comments.
3. Analyze and visualize the results. Detailed instructions about what should be shown are given in the write-up section.
    a. Show train and validation loss and report validation accuracy metric
    b. Plot the confidence calibration curve and report Expected Calibration Error (ECE)
4. Submit code and write-up on canvas.

Extra credit

Explore how different methods change the accuracy and calibration.

- Explore transfer learning strategies
- Simple regularizers / hyper-parameters like: L2 regularization / weight decay, dropout, early stopping, learning rate schedulers…
- More complex regularizers like Mix up training, stochastic weight averaging
- Data augmentation strategies: scaling, flips, color change, different image shapes, etc.
- Monte Carlo dropout (average test time predictions with dropout to estimate uncertainty)
- Use an ensemble of models, with simple averaging of the output probabilities.
- Bayesian neural networks, Bayesian model averaging

Code:

Template code is included, with TODOs of things to fill in. Comments also contain info about points and requirements. Please keep the function names for easier grading. (If you want to try a complex strategy that requires refactoring, just put a comment in the predefined function telling us where to look.) We prefer Ipython notebooks for the code (also easier if you’re running on Google Colab). On validation data, I got a cross entropy around 0.1 and accuracy around 81% after 50 epochs of training.

Write-up instructions:

The write-up should be as succinct as possible. Include the following sections, with the requirements in each section.

(a) **Hyper-parameter choices (1 point)** Report any hyper-parameter choices you made. For instance, learning rate and optimizer hyper-parameters, number of epochs of training, any preprocessing steps.
(b) **Plot train and validation loss curves (1 point)** Label your axes and curves.
(c) **Accuracy metric (1 point)** Report the accuracy on the validation set (averaged over all pixels and validation images).
(d) **Visualize an image from the validation set, its true segmentation, and the predicted segmentation (3 points)** See code for suggestions on better visualization
(e) **Plot confidence calibration curve on test data (2 points)** Show the confidence calibration plot.
(f) **Expected calibration error (1 point)** Report the expected calibration error
(g) **AI collaboration statement (1 point to include)**
AI "collaborators" are allowed (Github copilot, ChatGPT). I'm just curious how they were used and what worked.
(h) Extra credit (optional)

Collaboration:

Feel free to discuss ideas with classmates, but you should do the coding and write-up on your own. AI tools are allowed, please declare how they were used in the write-up.