HW2 Report

CSC-487 Deep Learning

Code Explanation

[!question] Briefly explain your solution and any design choices you made that weren’t specified in the instructions.

I normalized the images to [0,1] and reshaped to include a channel dimension, while keypoints were scaled using min/max values. I built a baseline VGG-style CNN and added a custom masked MAE loss to ignore missing keypoints. I then improved the model by adding Batch Normalization and integrating a learning rate scheduler to reduce overfitting.

[!question] Clearly describe any external sources that were used (e.g. websites or AI tools) and how they were used.

I referred to the PyTorch documentation for model building and training loop design and used online resources for implementing a learning rate scheduler and early stopping.

Discussion

[!question] What is the shape and data type of each provided matrix?

• **Images:** The image's shape is (7049,1, 96,96) as an int64 and then converted to float32 and normalized.

• **Keypoints:** The keypoints matrix has shape (7049, 30) in float32 after scaling, with NaN values for missing coordinates.

[!question] What do the rows and columns represent?

• **Images:** Each row corresponds to one facial image.

- **Keypoints:** Each row contains 15 (x, y) coordinate pairs for a face.

[!question] What are the ranges?

- **Images:** 0–255 raw, scaled to 0–1.

- **Keypoints:** Scaled to 0–1; converted back to pixel units using the original min/max when reporting error metrics.

[!question] How do you evaluate test performance and diagnose overfitting?

I evaluate performance using the custom masked MAE loss and a rough accuracy metric (keypoints guessed within a threshold error). I compare training loss to test loss. There is a significant gap which indicates overfitting. I also visually inspect keypoint overlays on test images to verify generalization and the original model before architecture changes do a little bit better.

[!question] Compare the baseline and improved models in terms of test performance and overfitting.

Just from visual inspection, the baseline model does better than the batch normalized model and the stop training loop. This can also be seen with my accuracy score and loss during epochs.