Leila Islam

MSc Dissertation Meeting

May 31, 2018

**Topics**

1. **Dissertation recap:** Confidence Calibration in Neural Networks through Data Augmentation via Latent Space Interpolation

This project will test whether augmenting training data using a generative model can improve NN calibration. The main steps are as follows:

* 1. Augment image datasets (use generative models to perform latent space interpolations)
  2. Train CNN on augmented dataset produced in the previous step
  3. Evaluate calibration of neural networks relative to those trained using non-interpolated images

1. **Rough timeline: hand in August 31 (~12 week duration)**
   1. Choose generative model to use for interpolations, CNN architectures that will be trained, and dataset (1 week)
   2. Augment datasets using chosen generative model (3 weeks)
   3. Train CNNs on augmented datasets (2 weeks)
   4. Interpret results, create plots, compare to other calibration methods (2 weeks)
   5. Finalize lit review, theory, methodology (4 weeks)
2. **Generative model to use for dataset augmentation**
   1. Adversarial autoencoder
   2. Wasserstein autoencoder
   3. *(Other options in table below)*
3. **Dataset(s)**
   1. MNIST for debugging
   2. CIFAR-10
4. **Latent space interpolations to consider**
   1. Simple linear interpolation
   2. Spherical linear interpolation (White, 2016)
   3. Interpolation done treating the latest space as Riemannian manifold (Chen et al., 2018)
   4. …?
5. **CNN architecture:** Recommended architecture? Could try one of LeNet/ResNet/DenseNet?
6. **Measuring calibration**(use same metrics as Weinberger et al., 2017)
   1. Reliability (accuracy vs. confidence)
   2. Expected calibration error (ECE)
   3. Maximum calibration error (MCE)
7. **Interpretation of results***:* compare to other calibration methods (temperature scaling, etc…) (Weinberger et al., 2017)
   1. Calibration results
   2. Reliability diagrams (accuracy vs. confidence)
   3. Computation time
   4. Ease of implementation
8. **Meetings:**
   1. Weekly on Thursdays unless otherwise specified?
   2. Dates travelling/at conferences?

|  |  |  |
| --- | --- | --- |
| **Generative model** | **Notes** | **Implemention found in?** |
| Adversarial Autoencoder (AAE) | Autoencoder trained with two objectives – reconstruction error criterion, as well as adversarial training criterion that matches aggregated posterior of latent representation to prior distribution. Sharp transitions in learned manifold (tightly packed) compared to VAE. | TensorFlow (MNIST) |
| Wasserstein Auto-Encoders (WAE) | Minimizes penalized form of Wasserstein distance between model distribution and target. Stable training and good latent manifold structure like VAEs, but better quality samples. | Tensorflow and PyTorch |
| Mutual Autoencoder | User can specify amount of information stored in latent code |  |
| PixelVAE | VAE model with autoregressive decoder based on PixelCNN. Less expensive than PixelCNN, latent codes learn non-trivial structure, but are more compressed than PixelCNN. | Tensorflow (MNIST) |