CS 165: Foundations of Machine Learning **Winter 2019**

Lecture 13: February 26, 2019

Lecturer: Anima Anandkumar *Scribe: Becky Roh, Michelangelo, Tim Krasnoperov*

13.1 Announcements

1. Final report due March 19, 2019
2. Final presentations in one session (like a workshop) on March 22
3. Make-up class February 27, 2019 in Annenberg 105 at 5 pm

13.2 Recap of Last Lecture

1. Robust deep learning
   (a) Machine learning predictions are *mostly* accurate but brittle.
   (b) Humans are more robust than these machines.
   (c) Simple perturbations, perceptible to human eye, can drastically change performance of machines. There have been many proposed algorithms for creating these attacks, both black-box and white-box, including gradient ascent and evolutionary approaches[1].

$$\max_{\delta \in \Delta} Loss(x + \delta, y; \theta) \tag{13.1}$$

   (d) Question to ask: can you train your classifier to be robust?

$$\min_{\theta} \sum_{x,y \in S} \max_{\delta \in \Delta} Loss(x + \delta, y; \theta) \tag{13.2}$$

      i. Create an adversarial example (or ensure one does not exist) then train a robust classifier.
      ii. For additive perturbations, use simple constraints.
      iii. What is the relevant perturbation one should care about?
      iv. In addition to accuracy, what other details does one care about (e.g. safety)? This can change the design.
      v. Ian Goodfellow and Christian Szegedy are the pioneers in this developing field, and their papers, (i.e. [2]) are great references for further interest.

   (e) The fast gradient sign method (FGSM)

$$\delta = \epsilon \cdot sign(\nabla_{\delta} Loss(x + \delta, y; \theta)) \tag{13.3}$$

      i. Update neural network based on worst-case loss (more local method).
      ii. For deep networks, image is non-convex, so it is not easy to optimize over it. Computational cost is a concern.

(f) Biggest drawback of adv. robust ML from standard ML is poor in generalization. There is large gap in adv. training and adv. evaluation.

  i. Over-fitting of training data

(g) Adv. robust generalization needs more data.

(h) Adv. examples are inherent to NN's because of their linearity[2][3]. These examples are likely to be present further away from training data points. Due to the curse of dimensionality, an image that is "close" to another to the human eye may actually be incredibly far in terms of Euclidean space.

13.3 Today's Lecture

Contributors to success: (i) algorithms, (ii) computation, and (iii) data. But, what really matters is data. Data is much harder to obtain. Today's lecture focuses on active learning.

1. Deep learning systems are data-hungry (e.g. C systems train on ImageNet at 1M+ images)
2. How to cope with scarce data:
   - (a) Data augmentation: (standard approach) take one example, perturb it, and make multiple examples
     - i. e.g. slightly clip image, rotate, add random noise.
   - (b) Semi-supervised learning: if there are unlabeled samples and few labeled samples, can they both be used?
     - i. Important question, since in reality data is more likely to be unlabelled than labelled, and labeling is usually costly.
     - ii. Autoencoder pretraining is a technique that allows a network to begin detecting useful features in unlabelled data using *undercomplete* networks. This way, the labelled data goes a further way.
   - (c) Transfer learning: researchers start with big pre-trained model and transfer
     - i. Transferring lower layers of CNN's, for example, is useful in avoiding to retrain the simple, "low-level" functionality of vision like edge and shape detection.
   - (d) Domain adaptation: sophisticated extension of transfer learning
   - (e) Active learning
3. Active learning
   - (a) For sequential experiemental design
     - i. Drug discovery: how does one automate something that is done through human intuition?
     - ii. Robotics and control
     - iii. Protein engineering
     - iv. Material science
   - (b) Design decision
     - i. Pool-based sampling
       - A. Latency getting answers and labels is intensive.
       - B. We have fixed sample of instances.

        C. This approach is the most standard because it is the easiest to implement.

    ii. Stream-based sampling

        A. Data is coming in a stream, decision is made on labeling at instant, can ignore if it is too expensive but at the cost of losing sample forever.

    iii. Membership query synthesis

        A. Asks for examples for certain specifications

(c) Other considerations

    i. Acquisition function: how to choose samples

        A. If there is a large set of images of cats, for example, and classifier is pretty certain, then it does not make sense to get more images of cats.

        B. If uncertain, use human judgment.

    ii. Number of queries per round: tradeoff between computation and accuracy

    iii. Fine-tuning vs. training from scratch between rounds

        A. How to incorporate both old and new samples

        B. If there is a large set of new samples, be careful of ignoring past samples.

        C. Danger of over-fitting earlier samples

        D. Computationally expensive to use everything, so stragetize incrementally

(d) How do you measure uncertainty?

    i. Baysian models

    ii. The source of errors matters: we look at decomposition of errors

        A. Aleatoric: not explained by data (having more data will not explain the error)

            - Homoscedastic: uniform for all datasets
            - Heteroscedastic: depends on data

        B. Epistemic: uncertainty due to model and limited data

(e) Standard neural networks are poorly calibrated and do not model epistemic uncertainty well

    i. DNN's tend to be overconfident and suffer from unknown unknowns (they do not know that they are making errors).

    ii. Deeper the network (more non-linear), the problem gets worse.

(f) How to obtain Bayesian uncertainty cheaply

    i. Dropout method (Gal 2017)

        A. Train with dropout, sample n independent dropout masks, make forward pass with each dropout mask, and assess confidence based on agreement.

        B. Evaluate with ensemble, rather than with a single model.

        C. We will have different results based on dropouts, then compare the agreements.

        D. Introduce additional stochasticity.

        E. Better confidence,even with very little training data

    ii. Bayes-by-back propogation (weight uncertainty)

        A. Prior distribution, sample from posterior, given mini-batch of data

        B. Challenge is the scale: we cannot choose arbitrary prior distribution.

        C. Simple variational approach, an approximation of full posterior.

        D. Use simple prior distribution (independent Gaussian).

    iii. Bayes by Backprop Principle

        A. Optimize the weights for conditional distribution of labels given inputs.

B. Bayesian setting: place prior upon weights and find the maximum posteriori (MAP) weights
C. Variational approximation to solve optimization problem because it is rare to use the full posterior
   - Operate on gradients instead of exact minimization
   - Why? The problem is bottleneck, so it is expensive.

References

[1] `https://arxiv.org/abs/1412.1897`

[2] `https://arxiv.org/abs/1412.6572v3`

[3] `https://arxiv.org/pdf/1312.6199.pdf`