Deep Learning in Statistics: Practical Challenges

Edgar Dobriban
Assistant Prof., Statistics and Computer Science
University of Pennsylvania

May 31, 2021

Deep Learning in Statistics: Practical Challenges

1. Challenge 1: How to run DL models?
2. Challenge 2: How to gain expertise?
3. Challenge 3: How to keep up with developments?

Challenge 1: How to run DL models?

1. Several options:, depending on size of problem & resources
   - 1.1 Free cloud computing: Colab, etc
   - 1.2 Local: personal computer (CPU/GPU), server
   - 1.3 Paid cloud
2. Software - fast evolving
   - 2.1 PyTorch, TensorFlow
   - 2.2 wrappers: JAX, Keras
   - 2.3 interfaces: Jupyter notebooks

Challenge 2: How to gain expertise?

1. Our short course
2. Online courses: Coursera's `deeplearning.ai`, Stanford Classes (CS231N, CS224N), DeepMind x UCL, ENS/Ecole Polytechnique [1]
3. In-person courses at university
4. Online materials: tutorials - search!
5. Many papers have GitHub repositories with code (search for popular repos, curated lists like `https://github.com/mbadry1/Top-Deep-Learning`)
6. Books?

[1] See resources collected at my course website: `https://github.com/dobriban/Topics-in-deep-learning`

Challenge 3: How to keep up with developments?

1. Tutorials: at conferences (NeurIPS, ICML, ICLR, AISTATS)
2. Attending conferences
3. Summer schools: video lectures on advanced topics (MLSS, EEML)
4. Review papers: LeCun et al 2015, Schmidthuber 2015; statistical: Yuan et al 2020, Fan et al 2021
5. Papers: influential papers (citation count), arxiv (firehose! need filter like `arvix-sanity.org`)
6. Other online sources for real-time updates: Twitter, blogs (Google, DeepMind)
7. Online seminar series: One World ML, DL: Classics and Trends
8. Theory: Programs at Simons Institute, IAS; course materials by leaders

Failure Modes in Deep Learning

Adapted from CIS 700: Deep Learning for Data Science, taught by Prof. Konrad Kording, David Rolnick and Jeffery Cheng

Failure Modes in Deep Learning

Symptom: Network won't go above random performance.

- **Cause:** Poor architecture (too narrow or deep), see here and here
  - **Diagnostic:** Increasing width or decreasing depth fixes issue
- **Cause:** Hasn't started learning yet
  - **Diagnostic:** Wait a couple epochs, plot performance and check significance.
- **Cause:** Poor initialization, see here
  - **Diagnostic:** Try other initializations. If using ReLUs, a good rule of thumb is (non-truncated) He normal initialization for the weights.
- **Cause:** Problem simply not solvable
  - **Diagnostic:** Would a human be able to solve it with lots of time? If not, good sign deep learning may fail. Also check completely different architectures and hyperparameters.
- **Cause:** Poor set of input features.
  - **Diagnostic:** Should the data be normalized? Might preprocessing the data in some way make the solution easier to find/express?
- **Cause**: Implementation bug (i.e., in your code).
  - **Diagnostic:** Can you overfit a small subset of the training data?

Failure Modes in Deep Learning

Symptom: Loss bounces over epochs.

- **Cause:** Learning rate too high
  - **Diagnostic:** Try decreasing the learning rate.

- **Cause:** Minibatch size is too small.
  - **Diagnostic:** Essentially equivalent to a large learning rate. Increasing the minibatch size or decreasing the learning rate should fix the problem.

Failure Modes in Deep Learning

Symptom: Catastrophic forgetting - i.e. learning how to solve a task and then forgetting it later in training.

- **Cause:** Data is presented sequentially instead of intermixed.
    - **Diagnostic:** Does shuffling the data fix the problem?

Failure Modes in Deep Learning

Symptom: Suspiciously good performance

- **Cause:** Encoded an answer in the input
  - **Diagnostic:** Look through input carefully and see whether cheating is possible.

- **Cause:** Test dataset too small
  - **Diagnostic:** Is there stochasticity in test accuracy between training runs?

Failure Modes in Deep Learning

Symptom: Plateaus at a strangely low performance.

- **Cause:** Learning rate too low.
  - **Diagnostic:** Try increasing the learning rate.
- **Cause:** Not using a good optimizer
  - **Diagnostic:** Try a different one, e.g. Adam instead of SGD. Can also change the hyperparameters within the optimizer.
- **Cause:** Too much regularization
  - **Diagnostic**: Try less regularization, for example, by decreasing the dropout proportion.