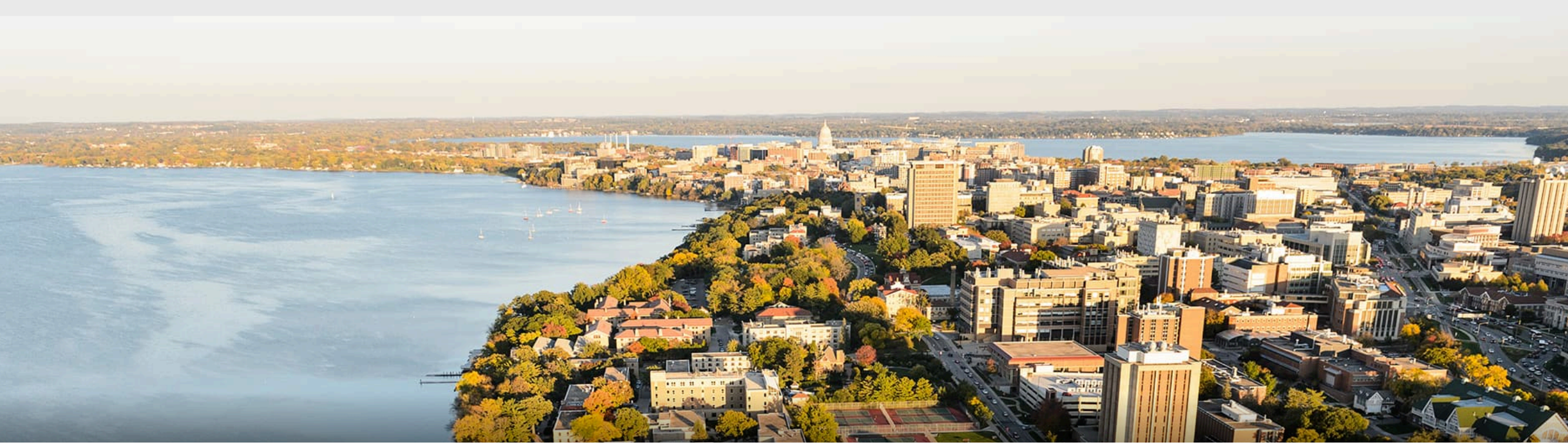


CS 760: Machine Learning
Practical Tips when Training Neural Networks

Kirthi Kandasamy

University of Wisconsin-Madison

March 8, 2023

Announcements

- **Midterm course evaluations.**

- **Homeworks:**
  - HW 4 due on Mar 20
  - HW 5 will be out early, due on Apr 3

- **Re-organizing lecture schedule** (will update website soon)

- **Midterm**:
  - 7:30 PM tomorrow (3/9/23). Will take 90 minutes.
  - B130, Van Vleck Hall
  - Arrive on time!
  - Bring scratch paper + cheat sheet + pen.

Training neural networks

Use existing DL libraries (PyTorch, Tensorflow etc.)
- Not worth implementing BP from scratch

First step: build a simple pipeline

- Set up data, model training, evaluation loop

- Overfit on one batch
  - Goal: see that we can get near-zero loss, catch any bugs

- Check that training loss reduces when you train.

Tips & Tricks: Data

- Shuffle the training data
  - In training ,usually don’t select random examples, but rather go through the dataset for each epoch
  - Shuffle to avoid relationships between consecutive points

- Pay attention to your data
  - Missing values, NANs, default values etc

Tips & Tricks: Initialization

Usually want to pick small random values to initialize weights

- Don’t want the same value: symmetry means every weights has same gradient, hard to break out of

- Multiple methods: various rules of thumb
  - Sample from a normal distribution
  - Note that #inputs affects the variance… grows as d^2 for d inputs. Helps to normalize, when initializing.

Tips & Tricks: Learning Rate Schedule

- Simple ways:
  - Constant
  - Divide by a factor every fixed number of epochs (annealing)
  - Look at training/validation loss and reduce on plateau

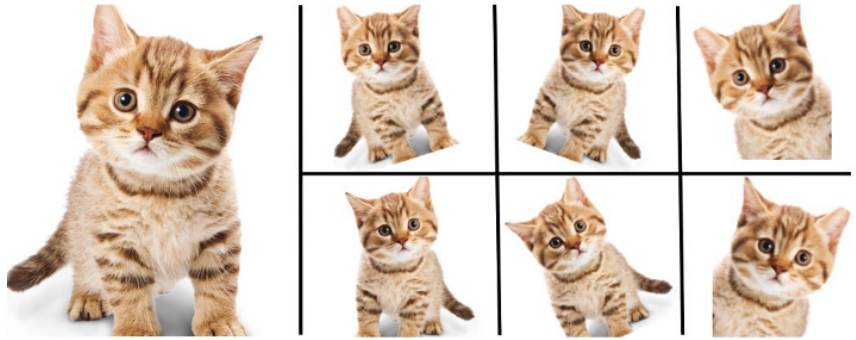
- Also simple: use an optimizer like Adam that internally tracks learning rates

- Lots of variations available

Tips & Tricks: Regularizing

- Best thing to do: get more data!
  - Not always possible or cheap, but start here.

- Augmentation
  - But make sure you understand the transformations



- Use other strategies: dropout, weight decay, early stopping
  - Check each strategy **one-at-a-time**

Tips & Tricks: Hyperparameter Tuning

Many solutions:

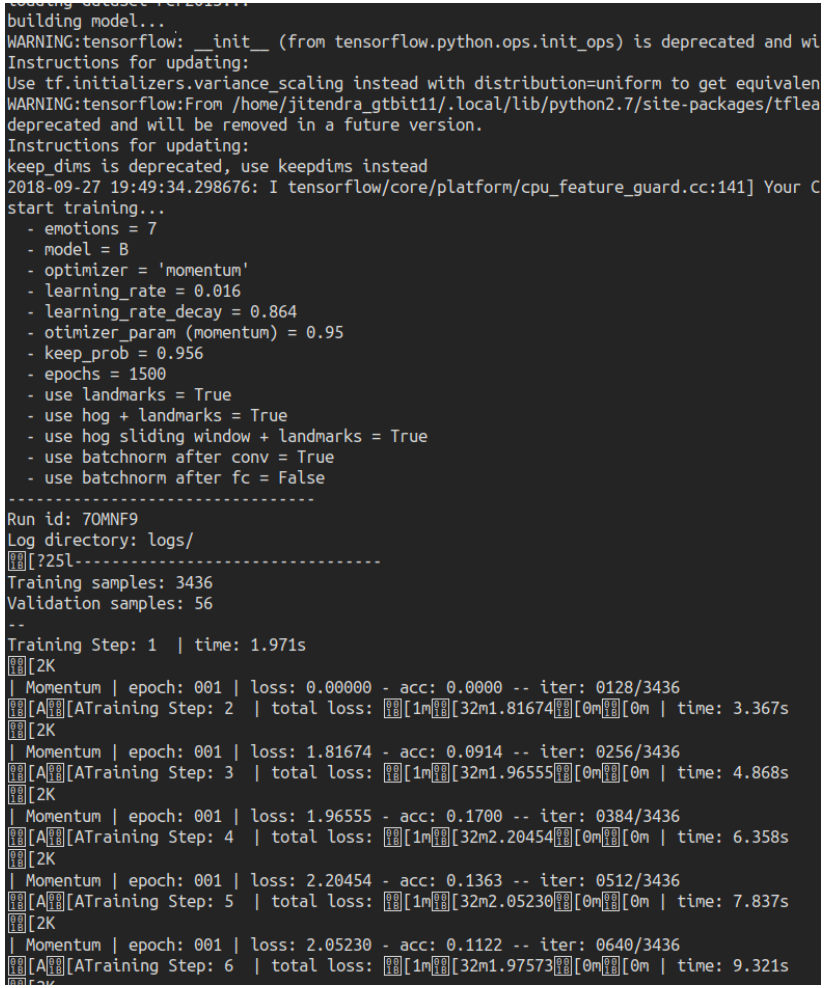
- **Grid Search**: pick candidate sets $S_1,\ldots,S_k$ for each hyperparamter, search over every combination in $S_1 \times S_2 \times \ldots \times S_k$
- **Random Search**
- **Advanced approaches:**

  Bayesian Optimization, Hyperband etc.

Tips & Tricks: Monitoring & Logging

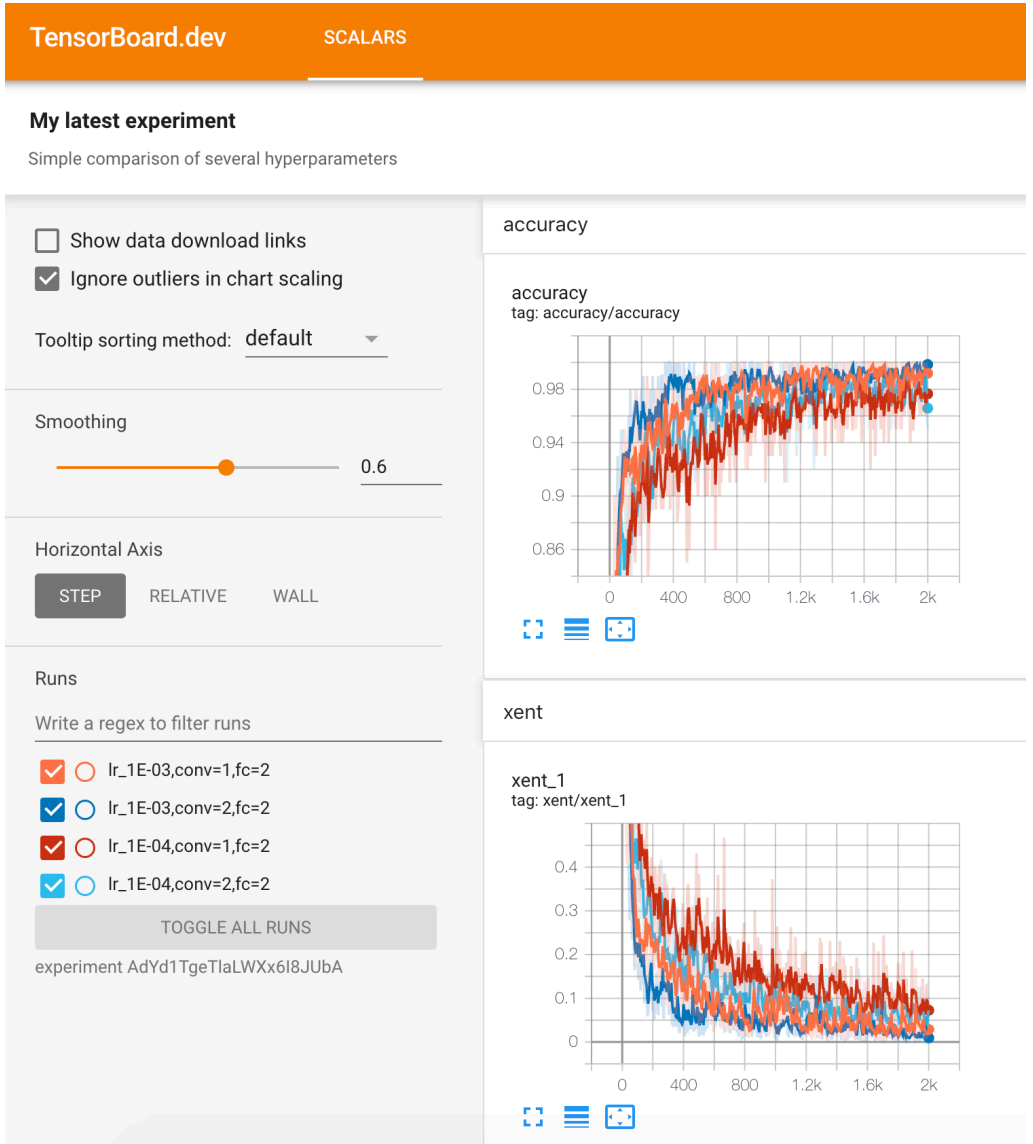
- Checkpoint your models (save weights) regularly!
  - Training can crash

- Log information from training process
  - Keep track of train / test losses, time elapsed, current training settings. Log regularly.
  - Can use this for early stopping as well.



Tips & Tricks: Monitoring & Logging

- Log information from training process
  - Use software packages
  - Also have built-in visualization
  - Example: TensorBoard



pytorch.org

Finally,

- You don’t always have to use the newest fanciest ML model.

- Sometimes simple models work well (esp in low data regimes)
  - E.g: Simple regression models with handcrafted features, kernel density estimation
  - Complex models may require expertise to get to work well.
  - Easier to interpret, incorporate domain expertise, and quantify uncertainty

- Incorporate domain expertise