Training neural networks: a practical guide for debugging

Adapted from slides by Dmytro Mishkin

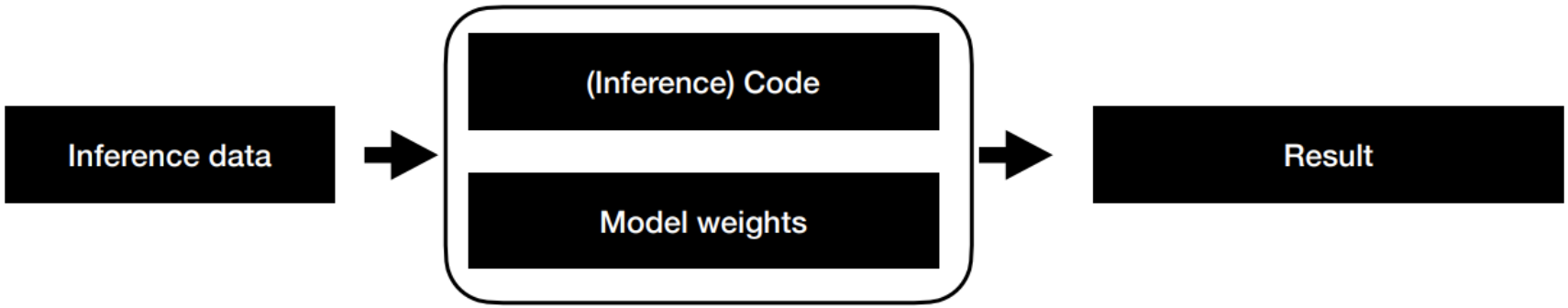
Debugging handcrafted code (RECAP)

1. Did I prepared minimal input and expected output? Math-based, or reliable library based
2. Did I visualize everything?
3. Did I print shape, data types, and values?
4. Did I check for a stupid mistakes? Like typos in variable names, naming variables as function
5. Did I check library versions and updates?
   E.g. old torch.solve(B, A), but torch.linalg.solve(A, B)
6. Do I have NaN-prone operations?
   e.g. log, sqrt, division, etc. Use eps there or some kind of guards
7. Do I have some memory sharing?
8. Is there anything hardcoded?
9. Can the bug in one function be compensated by other bug in other function?

Neural networks debugging

Most common issue: They fail silently!

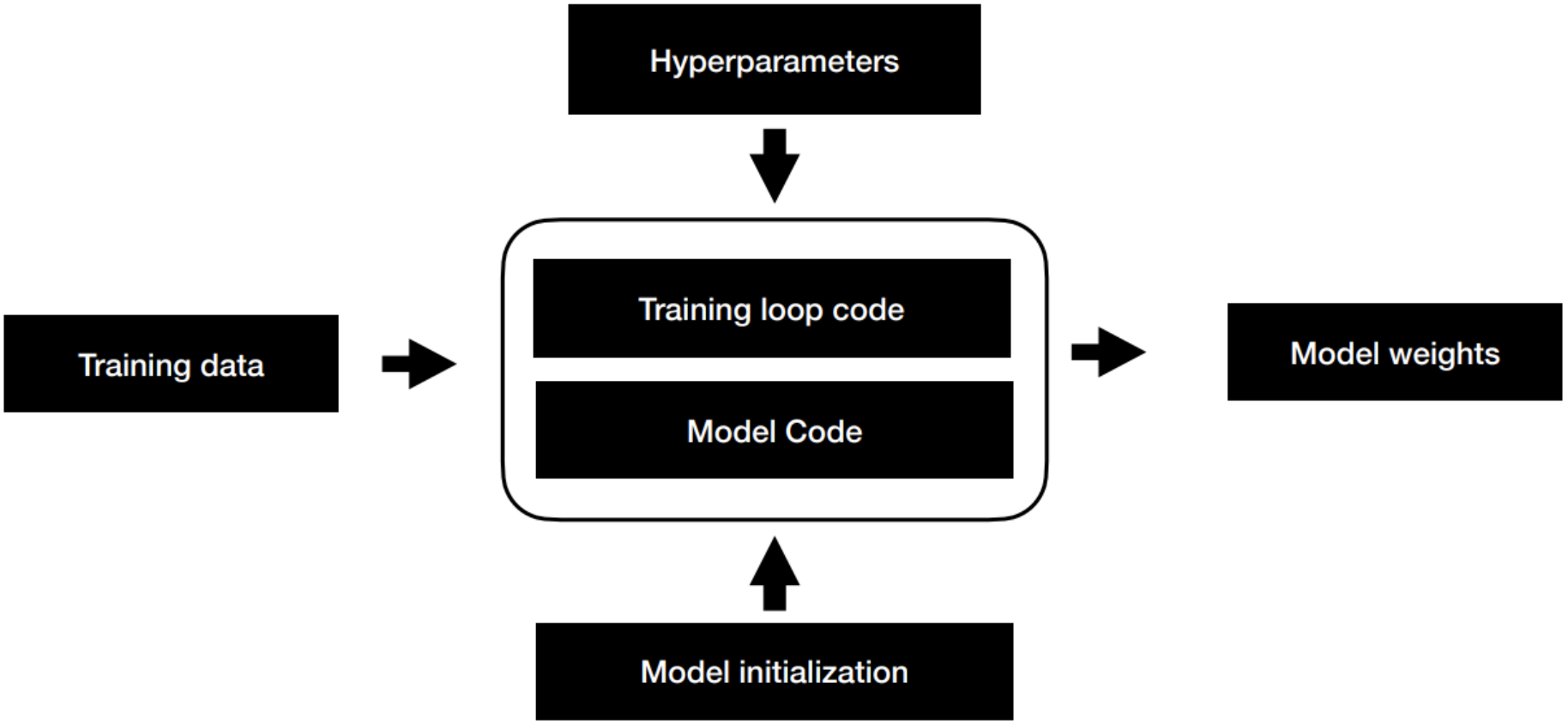
Neural networks inference (test-time)



Common issues with running a trained model

1. Not loading the weights
2. Not switching to .eval() mode
3. Applying wrong preprocessing

Neural networks training



Andrej Karpathy recipe

1. Become one with the data
2. Set up the end-to-end training/evaluation skeleton + get dumb baselines
3. Overfit to a single batch
4. Regularise

http://karpathy.github.io/2019/04/25/recipe/

Andrej Karpathy recipe: Overfit

1. Fix random seed
2. Disable any unnecessary code
3. Visualize inputs right before the net
4. Verify loss @ init
5. Train model with empty inputs
6. Verify decreasing training loss
7. Visualize prediction dynamics on single test batch
8. Use Adam optimizer

Andrej Karpathy recipe: Regularise

1. Get more data
2. Add data augmentation
3. Pretrain
4. Try different model size
5. Dropout, weight decay

Common issues with training

1. Not switching to .train() mode
2. Not applying proper data preprocessing
3. Not shuffling the data (esp. for BatchNorm)
4. Wrong learning rate/hyper parameters
5. Bad initialization