DS 207– Applied Machine Learning

Cornelia Paulik, PhD

School of Information

UC Berkeley

Hyperparameter Tuning in ML

Learning objectives

- Hyperparameter tuning definition
- Tuning techniques
- Monitor learning curves
- Debugging learning curves

Definition

- **Hyperparameter tuning** is the process of optimizing the hyperparameters of a ML model to improve its performance.

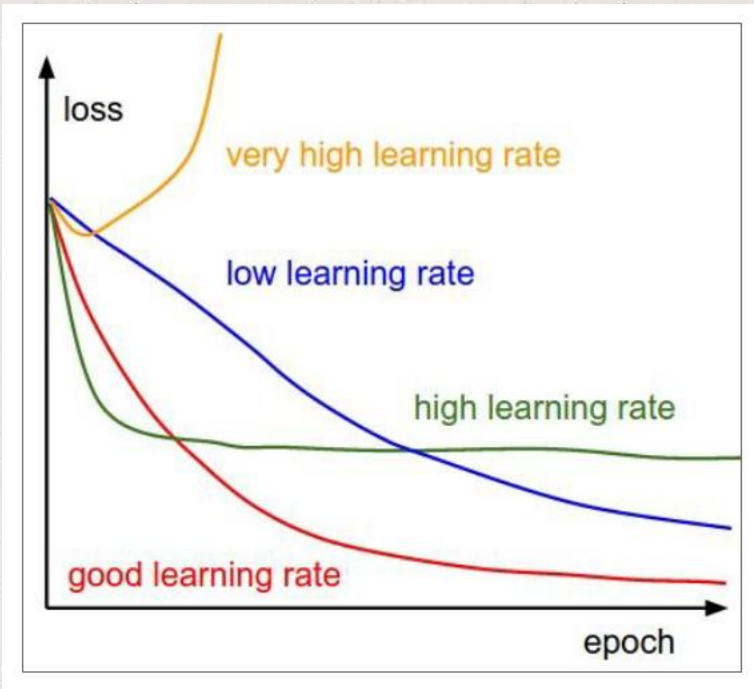
- Hyperparameter examples include:
  - learning rate
  - # of epochs
  - batch size
  - # of layers and units
  - optimizer

- Not the same thing as model parameters, which are learned during training (e.g., weights)

Tuning techniques

- **Manual search**: trial-and-error approach; you manually adjust hyperparameters based on experience and intuition
- **Grid search:** exhaustive search over a specified parameter space (can be computationally expensive)
- **Random search:** random sampling and evaluation of hyperparameter combinations (more efficient than grid search if the param space is large)
- **Hyperband tuning**: A novel bandit-based approach to hyperparameter optimization (very useful for deep learning)
- Others…

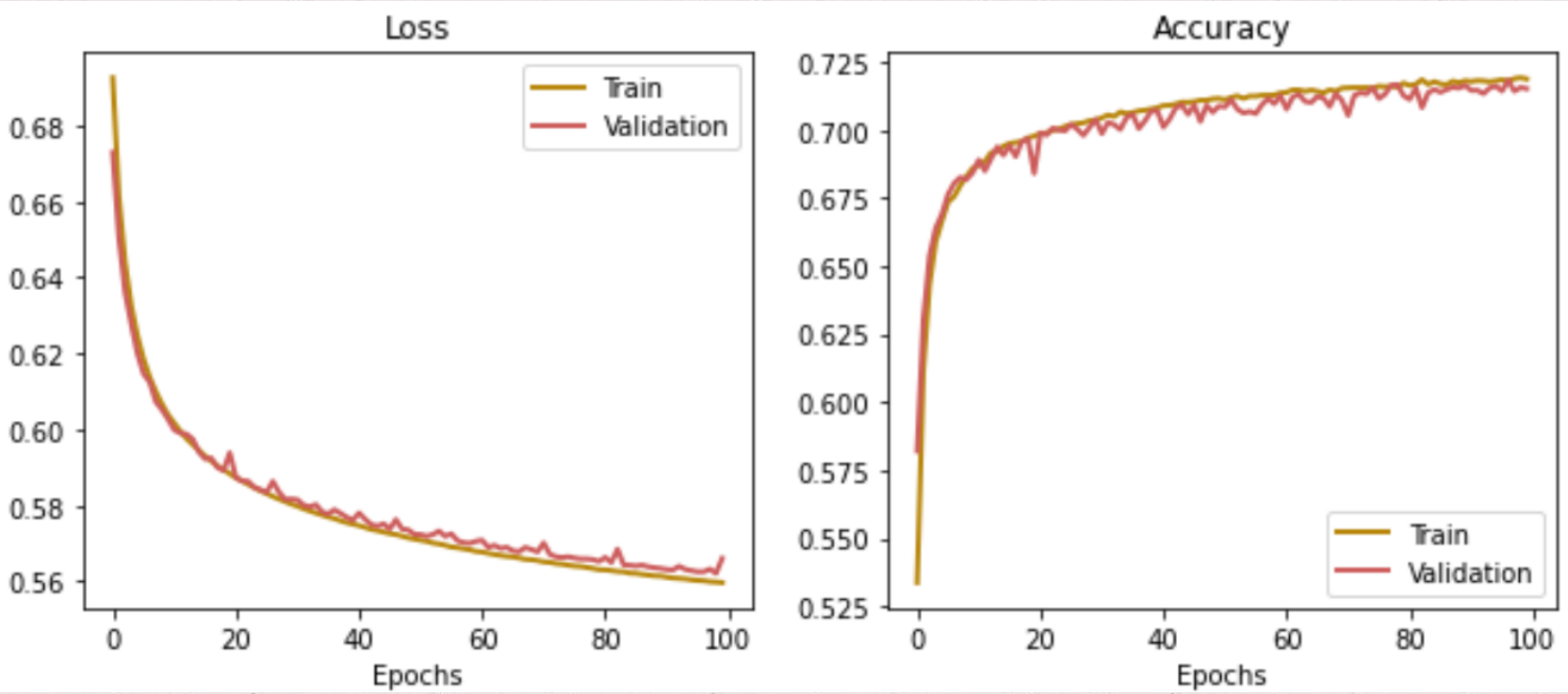
e.g., Best learning rate?



They are all good! They give you a hint on how to improve your model performance by adjusting the hyperparameter value.

Monitor learning curves

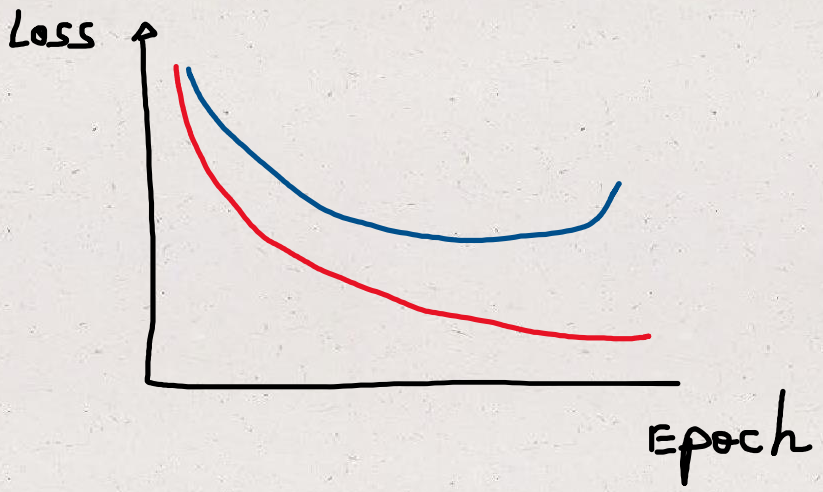
Monitor training and validation metric



Monitor training and validation loss

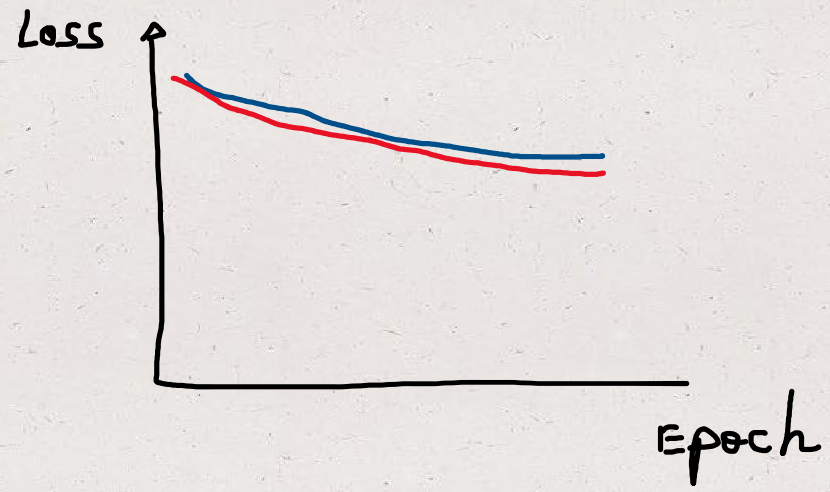
Debugging learning curves

Overfitting

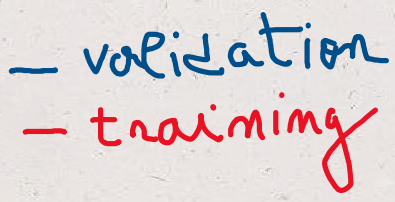


Training loss may continue to decrease but validation may get worse -> fix: stop training earlier

Underfitting

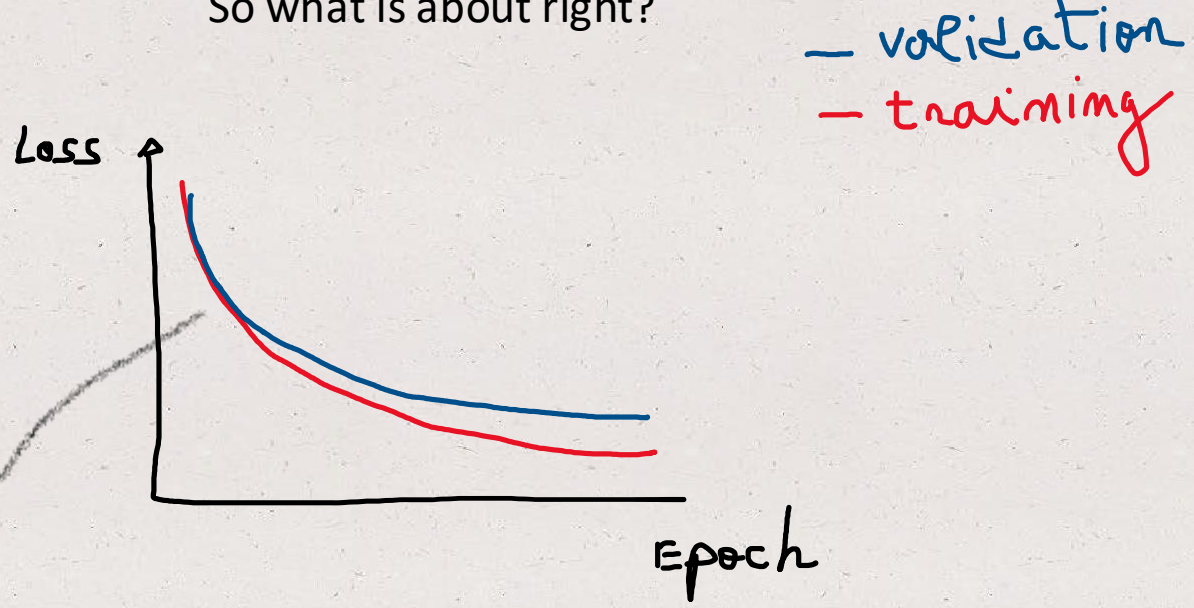


High loss, the model is not learning sufficiently. Small or no gap between training and validation loss -> fix: increase learning rate, reinitialize weights



Debugging learning curves

So what is about right?



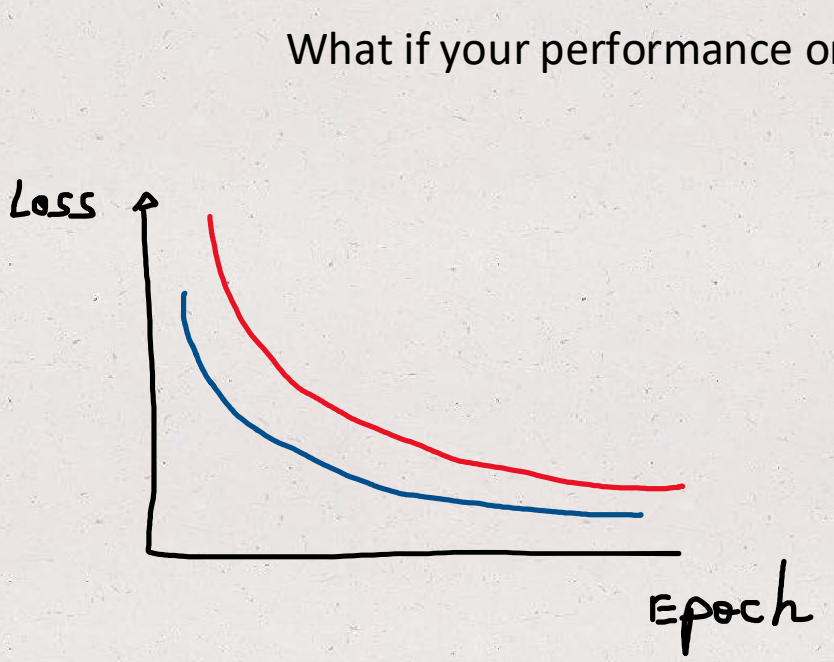
Best models overfit slightly (i.e., you want a little bit of gap between training and validation loss)

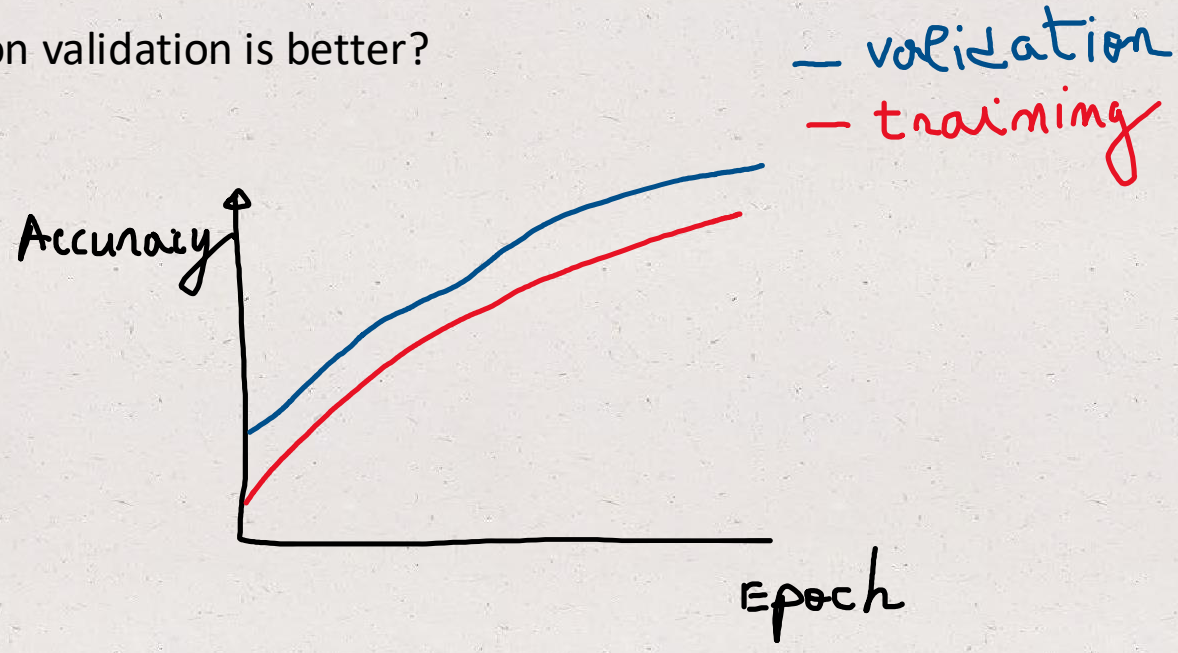
Why you want a gap: can improve generalization performance

Very nice elbow shape: notice the steep improvement at the beginning

Debugging learning curves

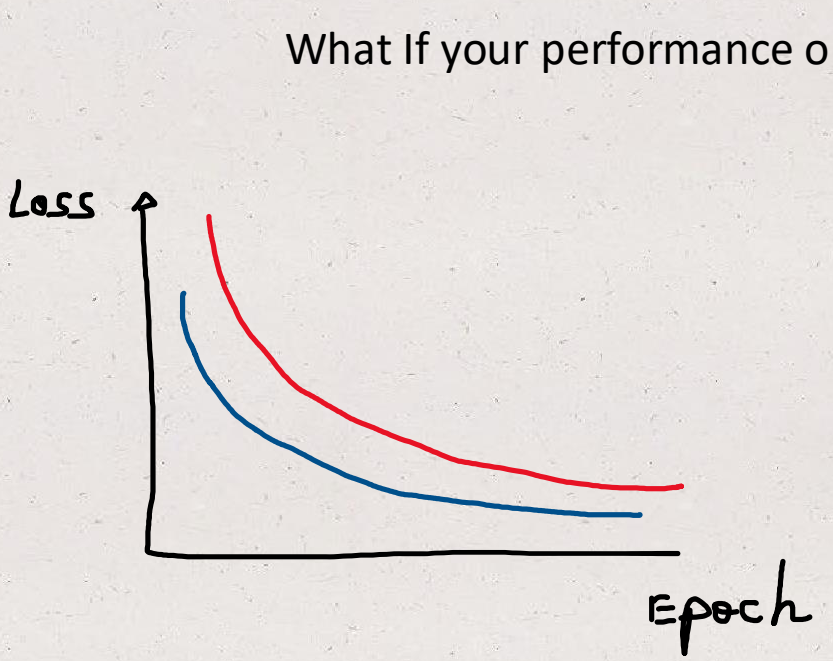
What if your performance on validation is better?

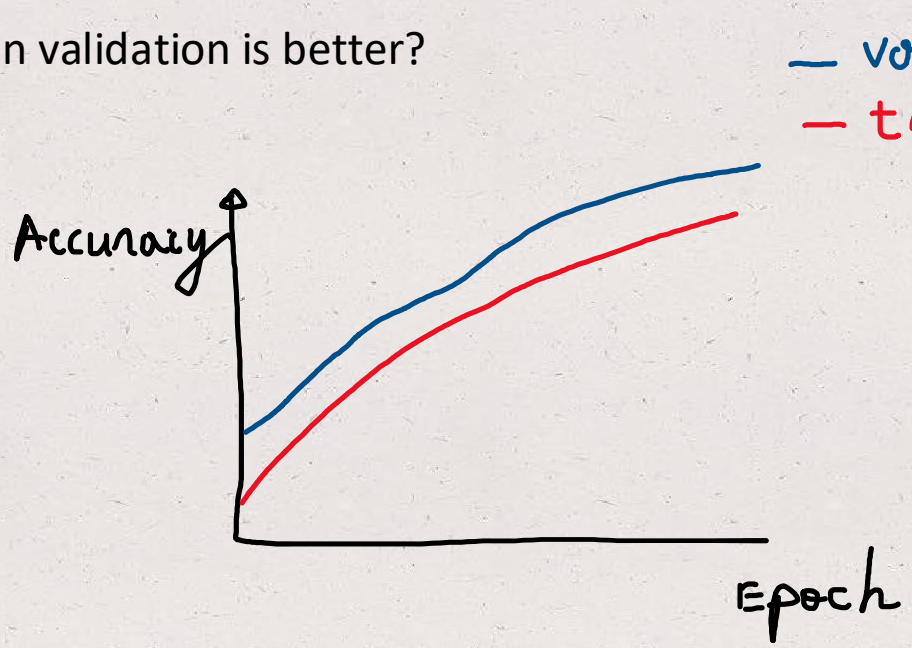




Debugging learning curves

What If your performance on validation is better?

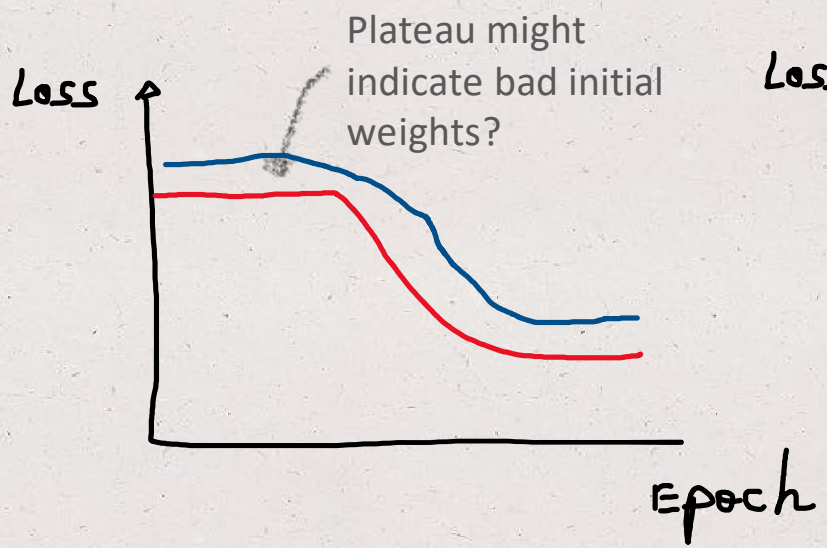




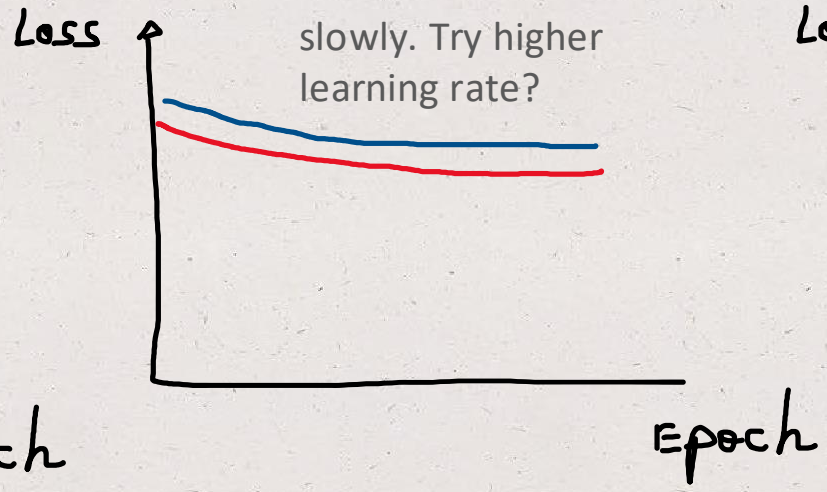
Your training data is likely underfitting. Even though this happens, your validation data may perform well under the situation.

Maybe your validation examples are different/easier to predict? Shuffle the data multiple times and average the prediction results.

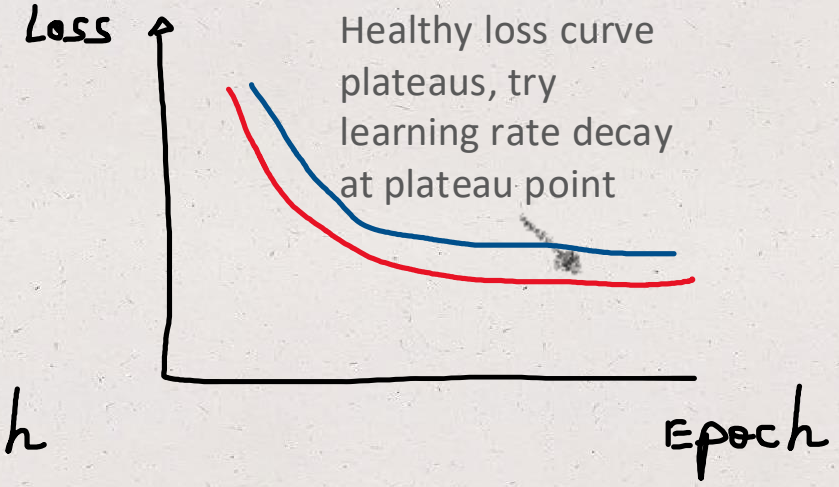
Debugging learning curves



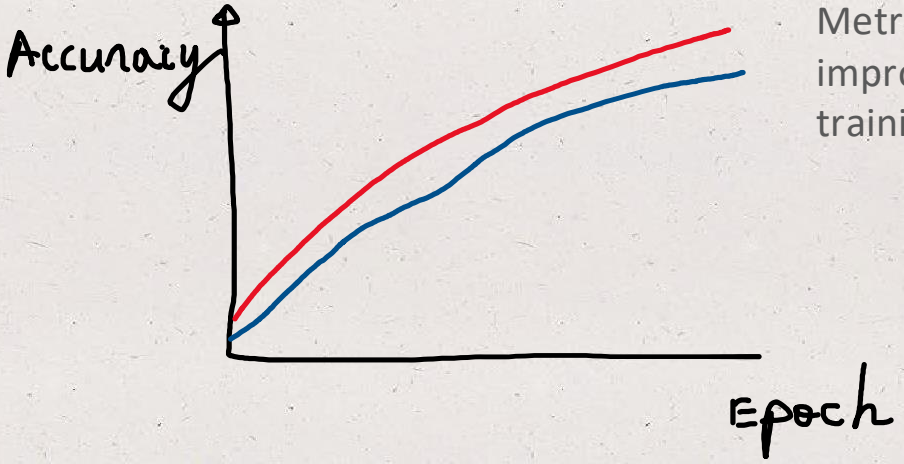
Plateau might indicate bad initial weights?



Loss decreases slowly. Try higher learning rate?



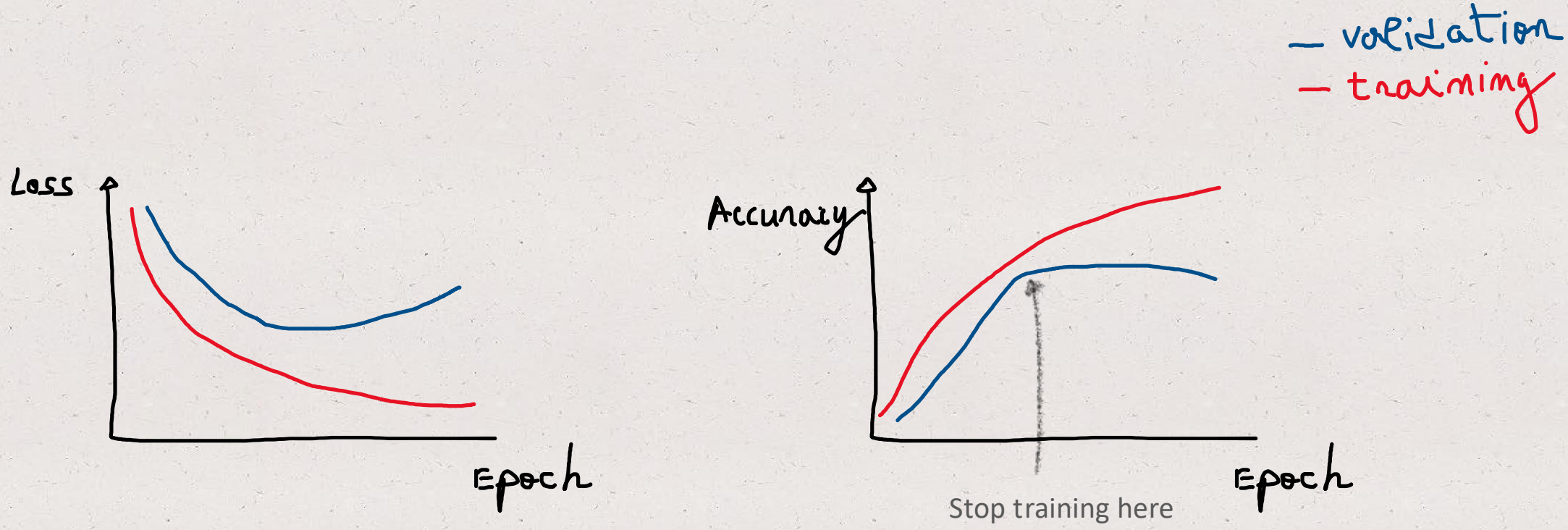
Healthy loss curve plateaus, try learning rate decay at plateau point



Metric still improving, keep training!

– validation
– training

Debugging learning curves

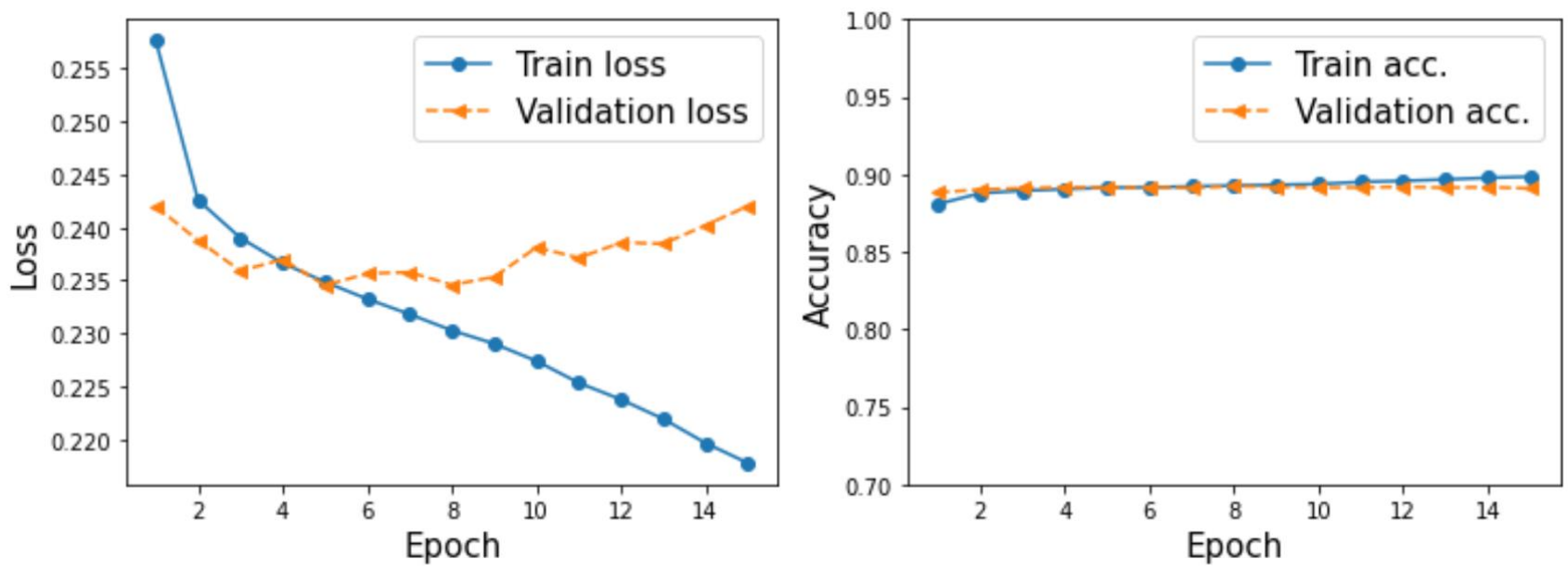


Early stopping:

stop training when the performance gap between training and validation increases!

Debugging learning curves

What do you recommend?



Debugging learning curves

What do you recommend?