

# W207— Applied Machine Learning

Cornelia Ilin, PhD

School of Information

UC Berkeley

Network Architecture and ML debugging



# Announcements

- Assignment 10 due this Sunday (this is the last one!)
- Don't worry about exact numbers for Exercise 4. The idea is to do an ablation study.




# Last week

- Sequential modelling
- Embeddings
- CNN for 1D data
- Application: Sentiment analysis based on drug reviews



# Today's learning objectives

- Ingredients for ML/AI success
  - How to debug learning curves
  - End-to-end application using my research as a motivating example
  - Short intro to RNN/LSTM (RM, Ch 16; will come back here in week 13)
- 

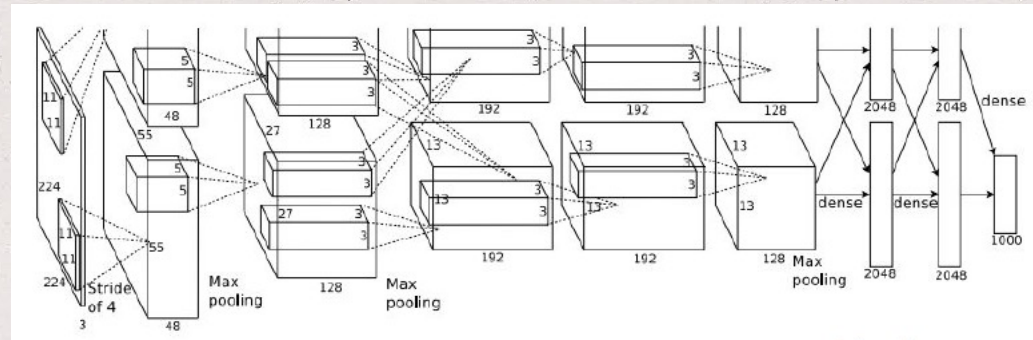


# Ingredients for ML/AI success



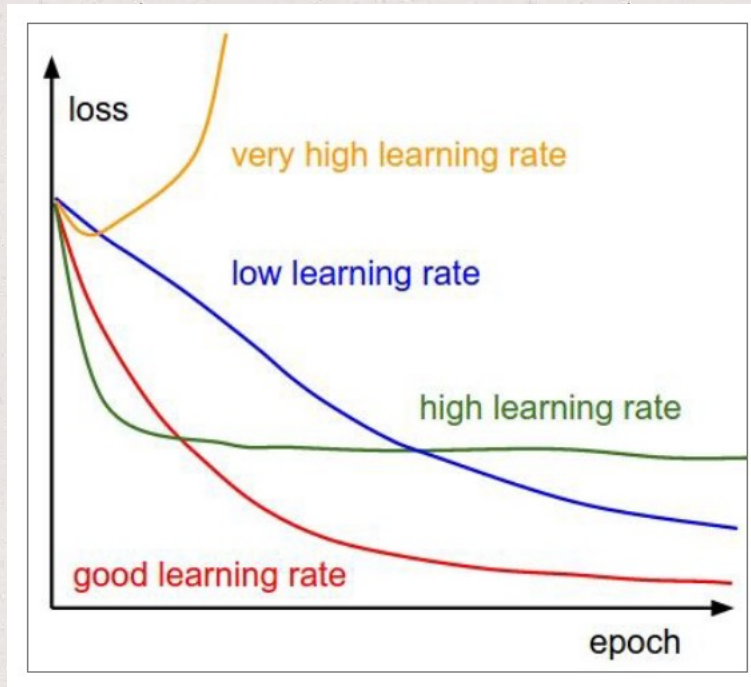
# Ingredients for ML/AI success

- Data →
- Algorithms
- Compute →





# Best learning rate?



→ They are all good!

Start with large rate and decay over time

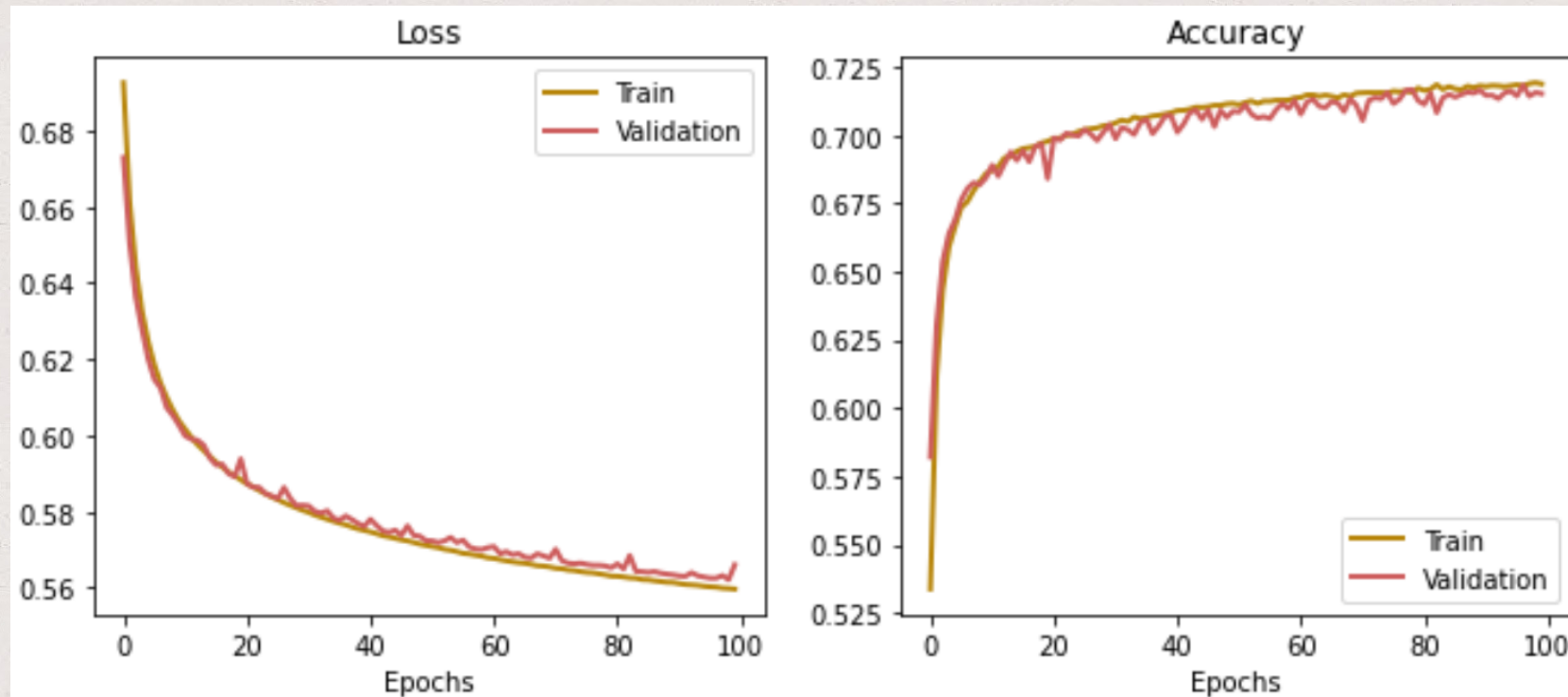
- at fixed points (20, 50, 80 epoch)
- cosine
- linear
- inverse sqrt

} *fancy schedulers*



# Monitor learning curves

Important to monitor final  
validation metric as well!

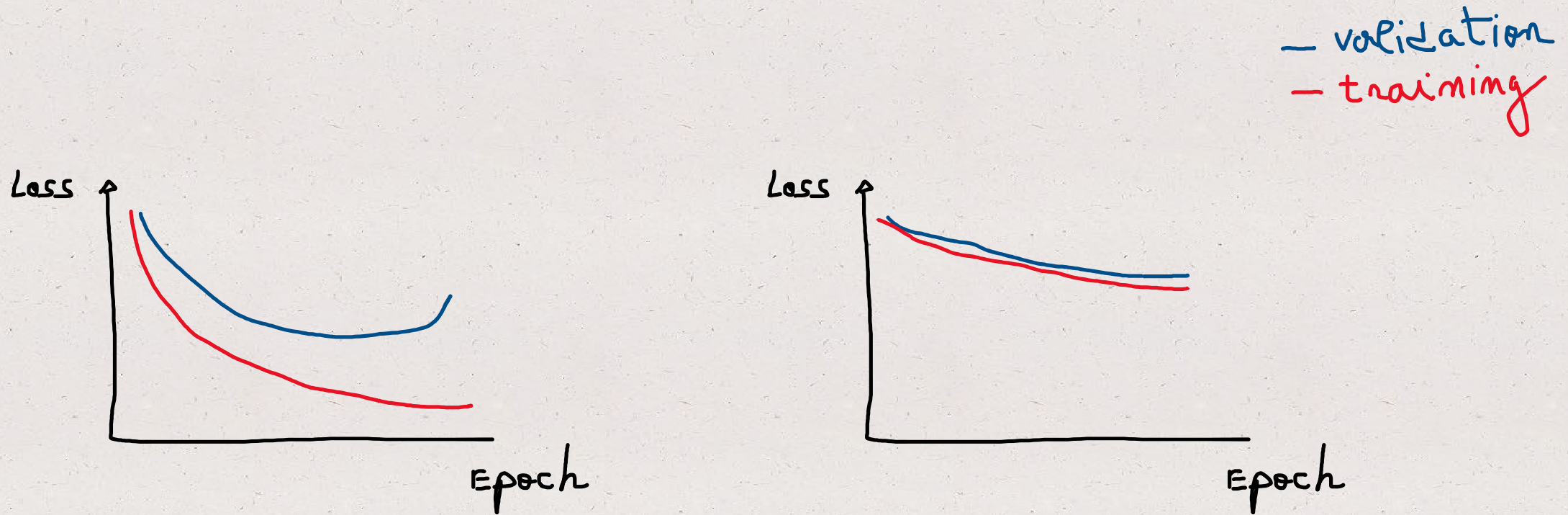


Evaluate validation loss





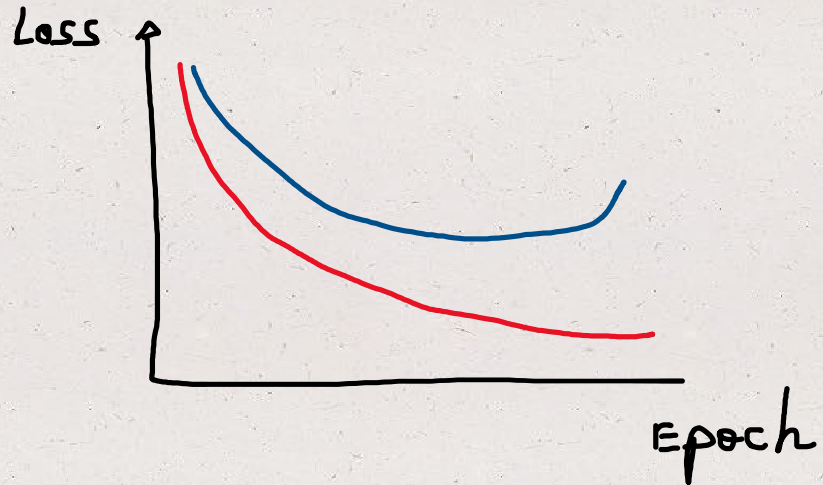
# Debugging learning curves





# Debugging learning curves

Overfitting

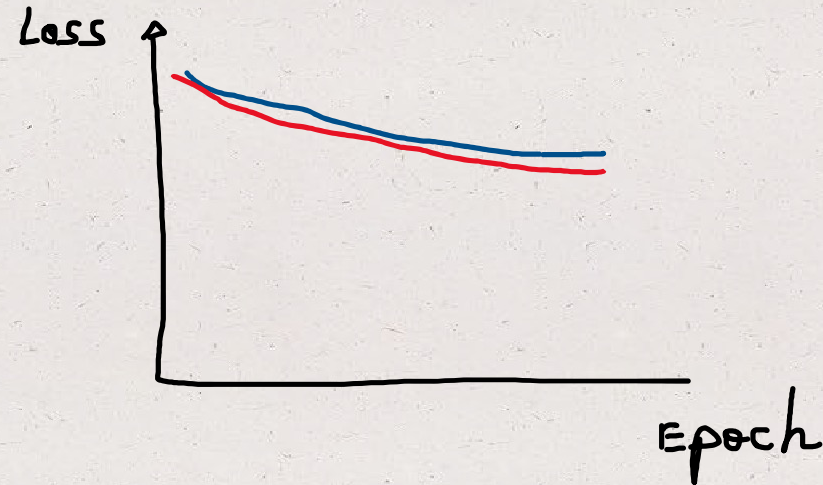


Training loss may continue to decrease but validation may get worse

How to improve:

- Increase data or regularize model, or
- decrease model capacity (make it simpler)

Underfitting



Small or no gap between training and validation loss. Model not learning sufficiently

How to improve:

- increase model complexity
- make task easier, clean data better

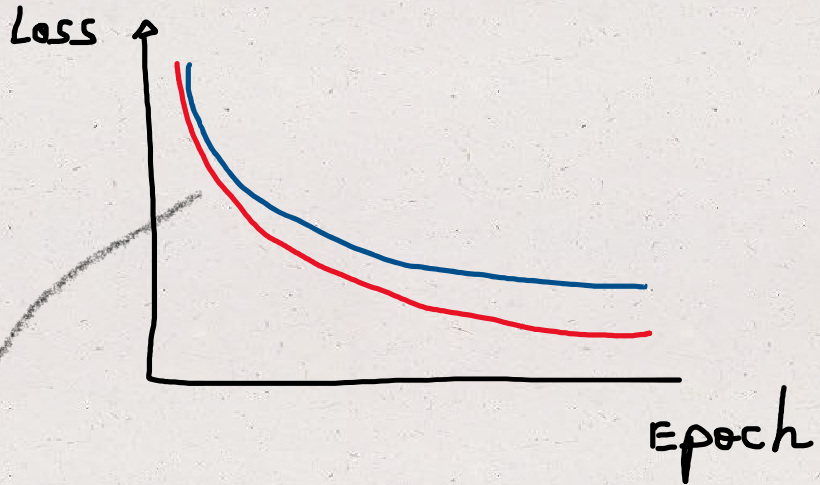
— validation  
— training



# Debugging learning curves

About right!

— validation  
— training



Best models often have small overfitting!

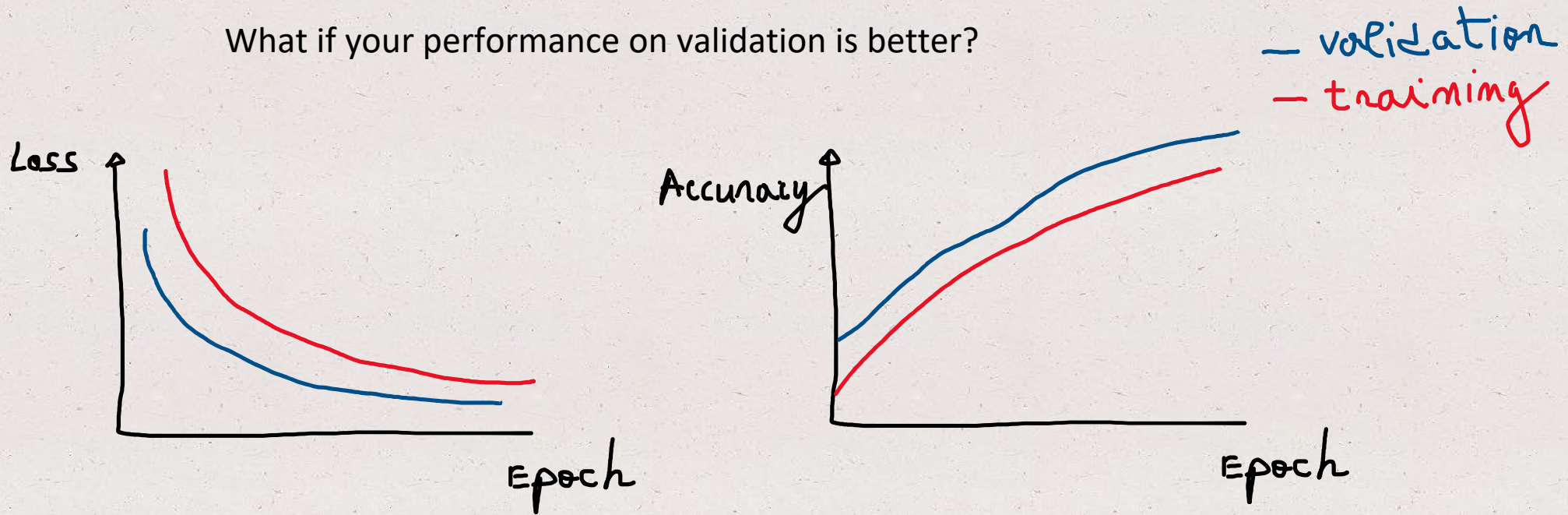
Push complexity of your model to the highest your data can handle!

Notice the steep improvement at the beginning



# Debugging learning curves

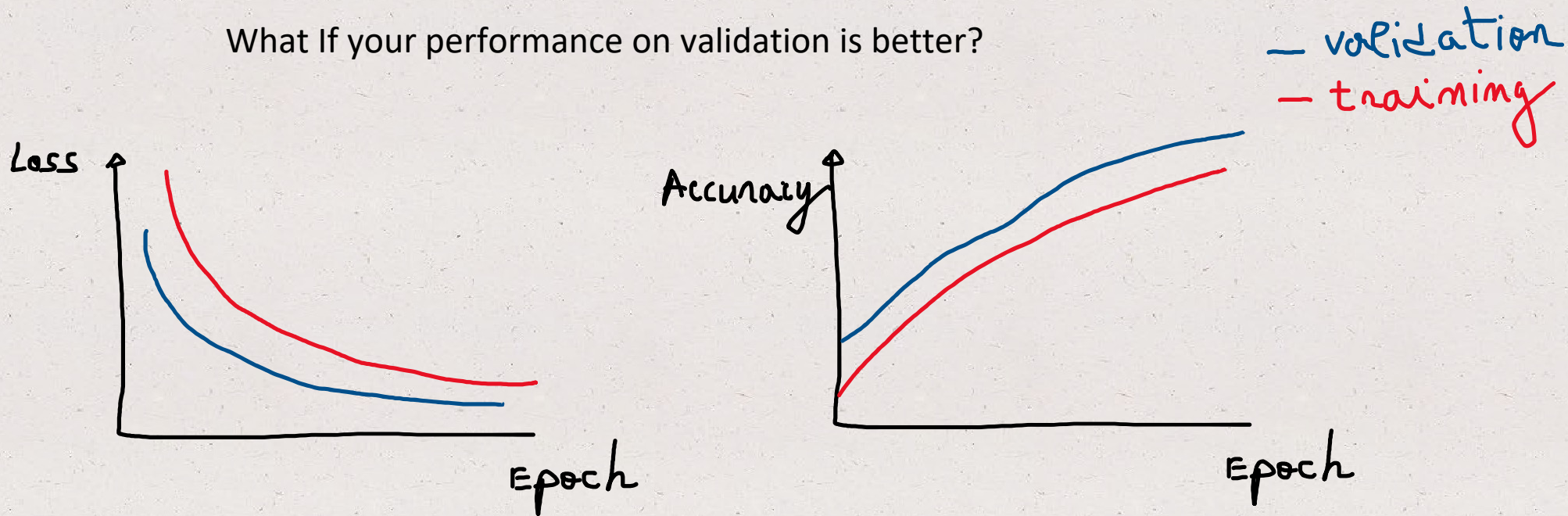
What if your performance on validation is better?





# 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 you have regularized too much? or

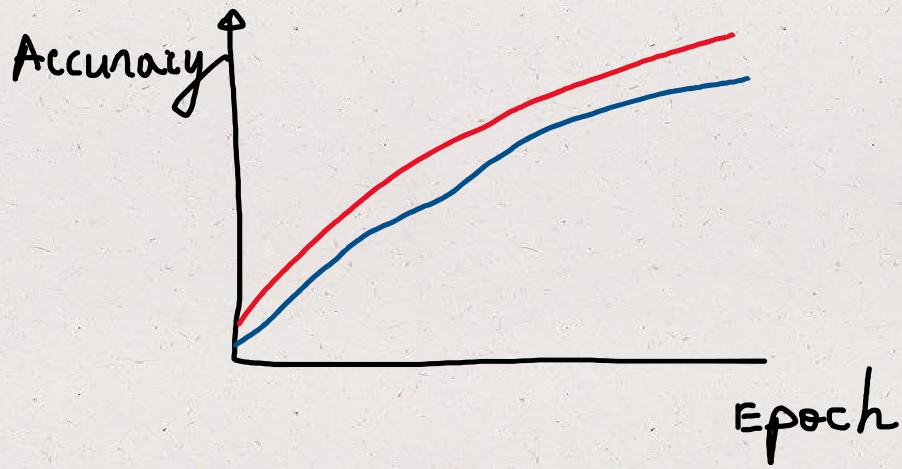
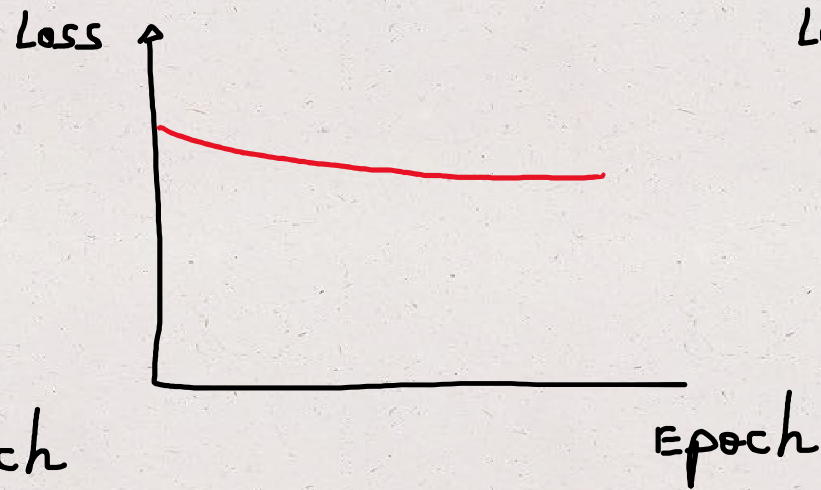
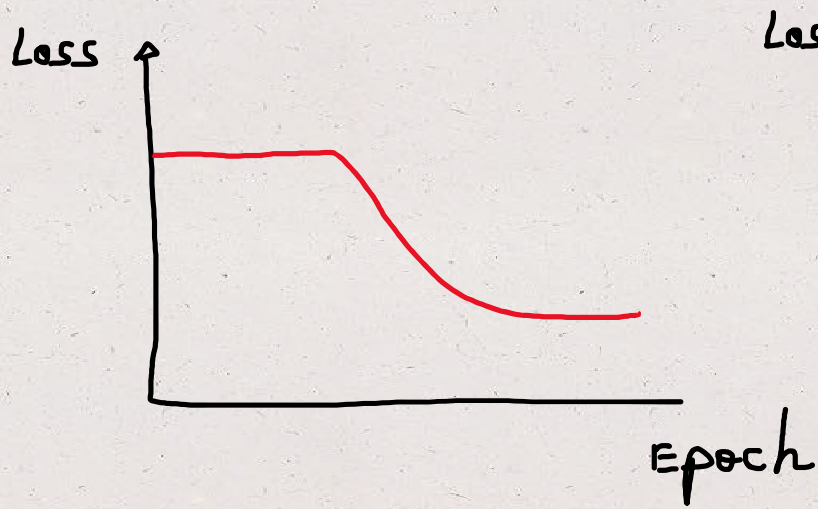
Maybe your validation examples are different/easier to predict?

How to improve:

- Increase model capacity (add extra layers)
- decrease regularization (e.g., number of Dropout layers or percentage of units dropped out)
- K-fold cross validation exercise



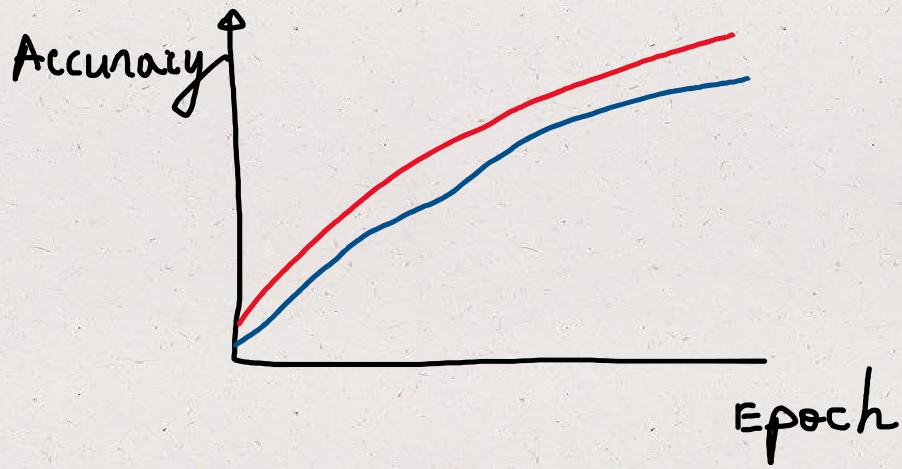
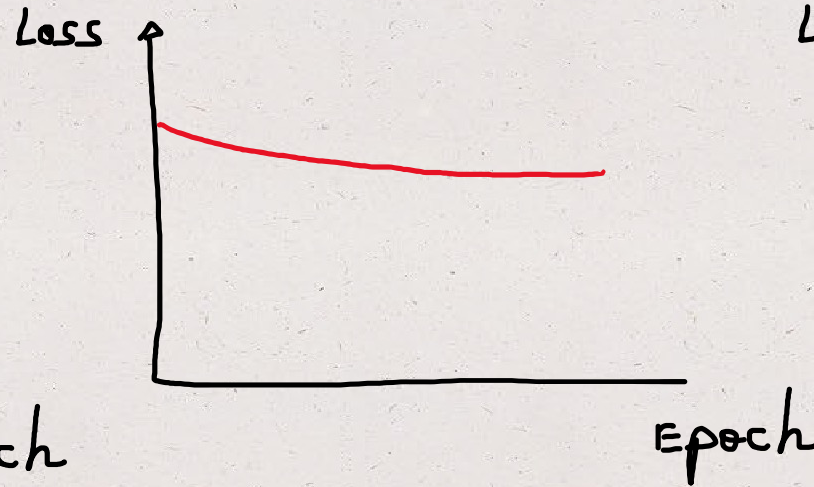
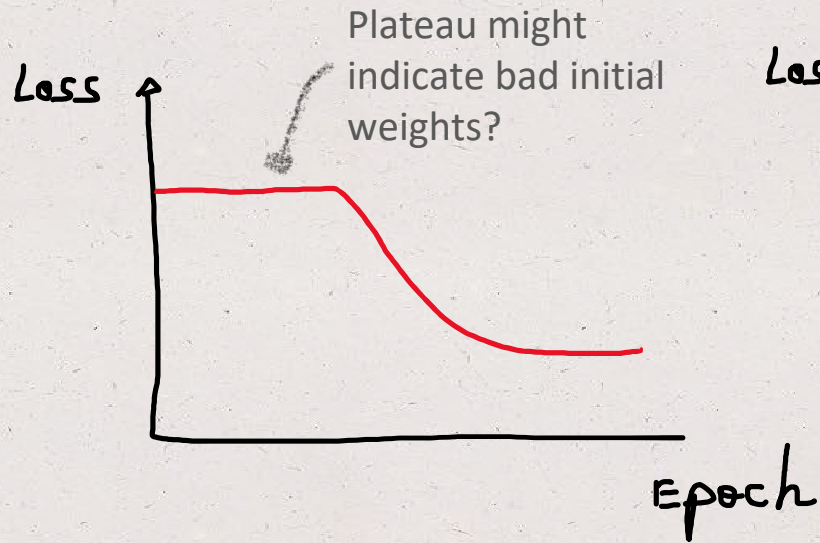
# Debugging learning curves



— validation  
— training



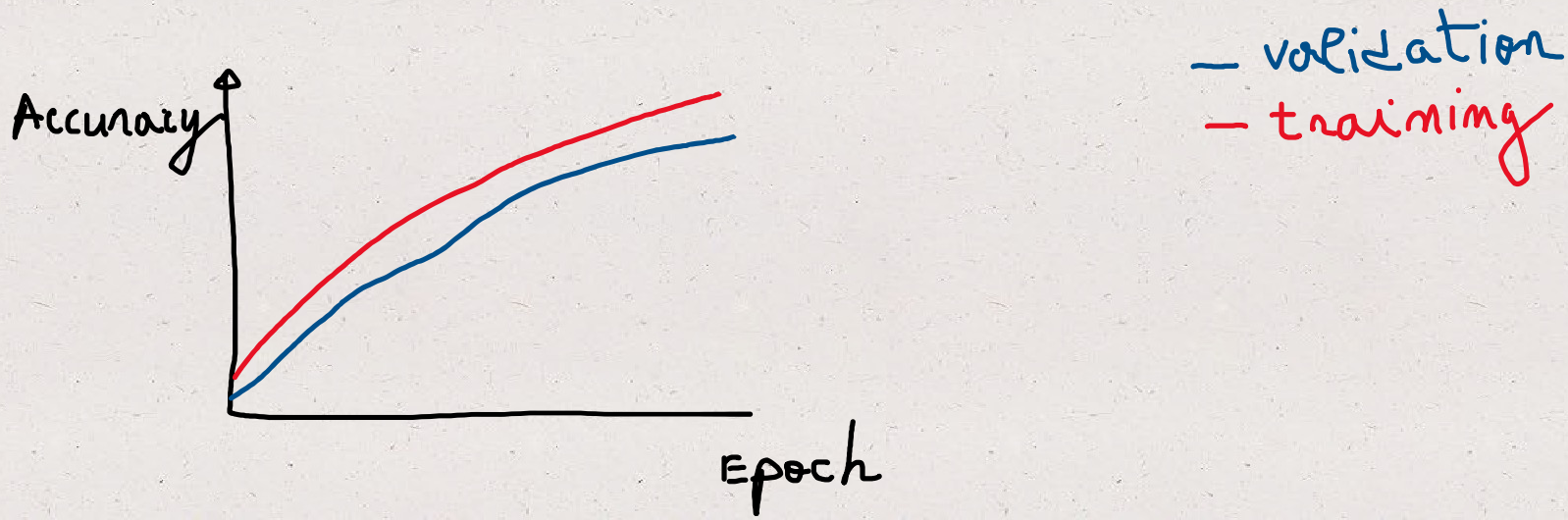
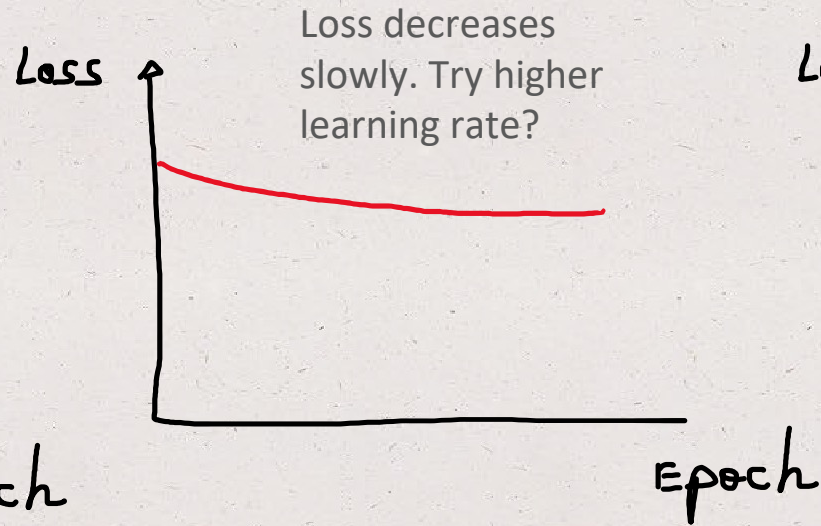
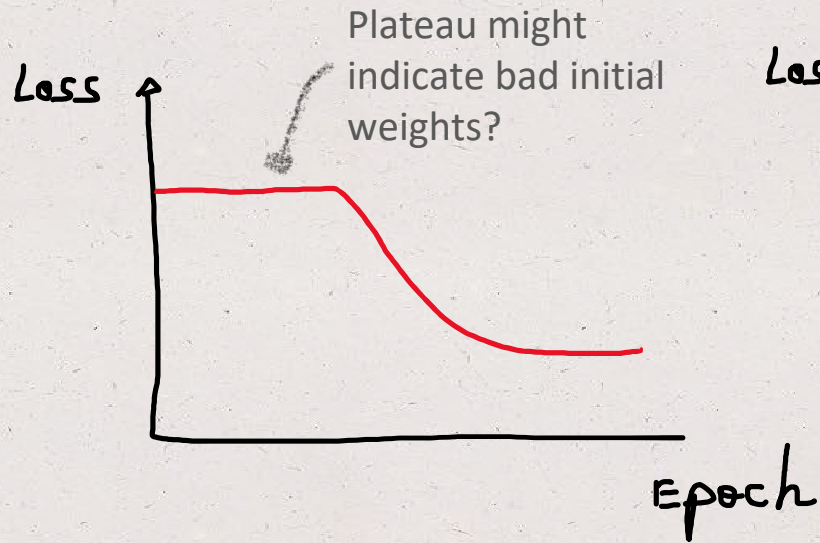
# Debugging learning curves



— validation  
— training

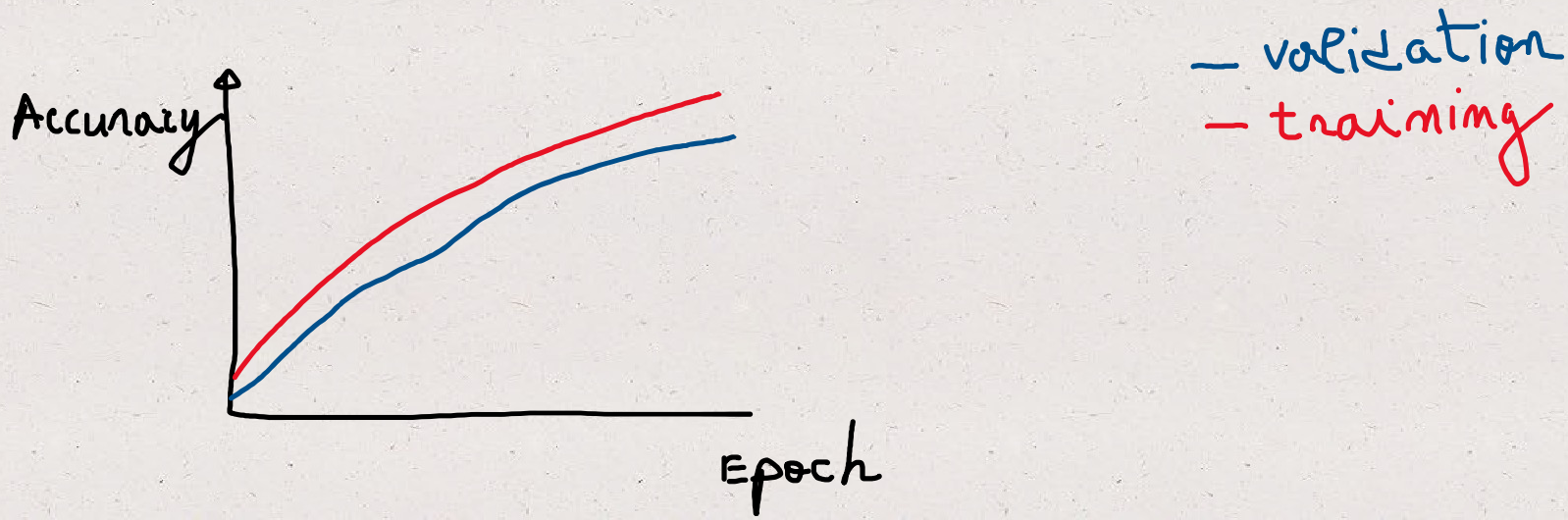
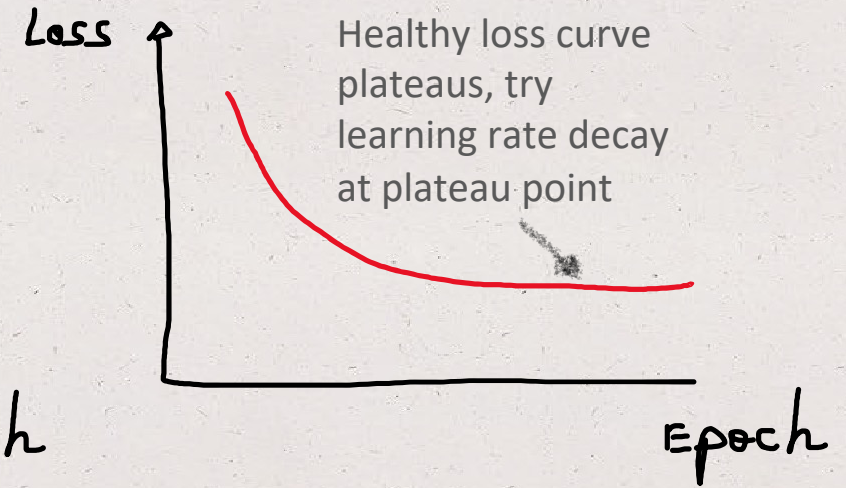
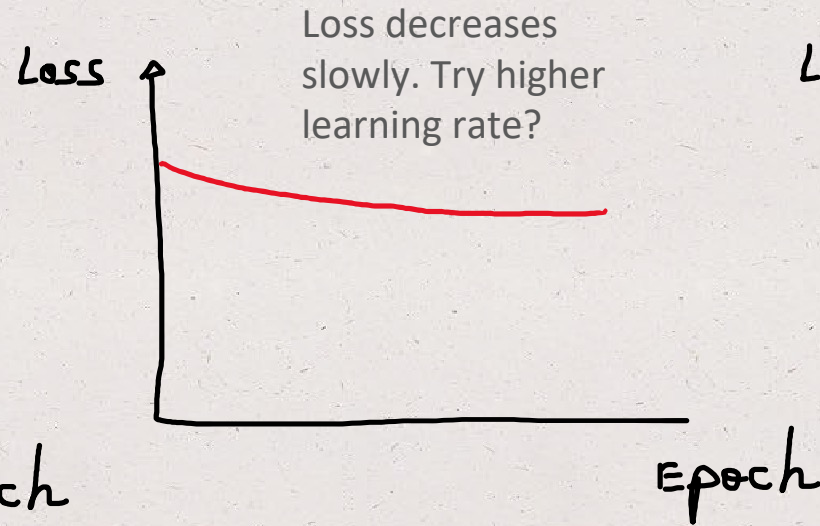
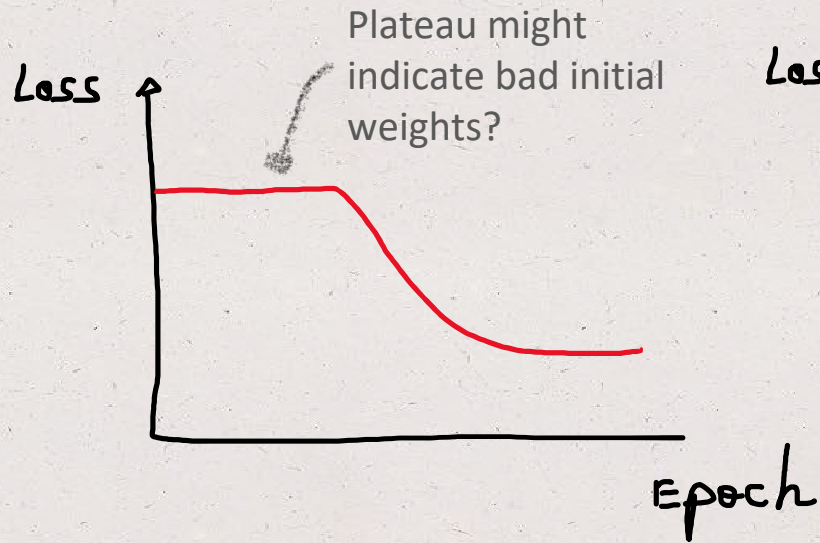


# Debugging learning curves



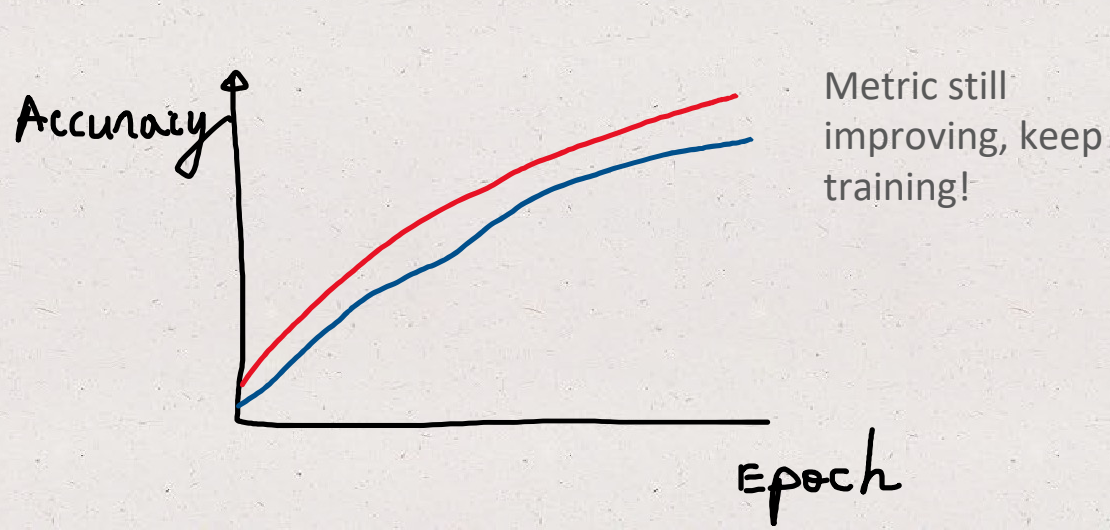
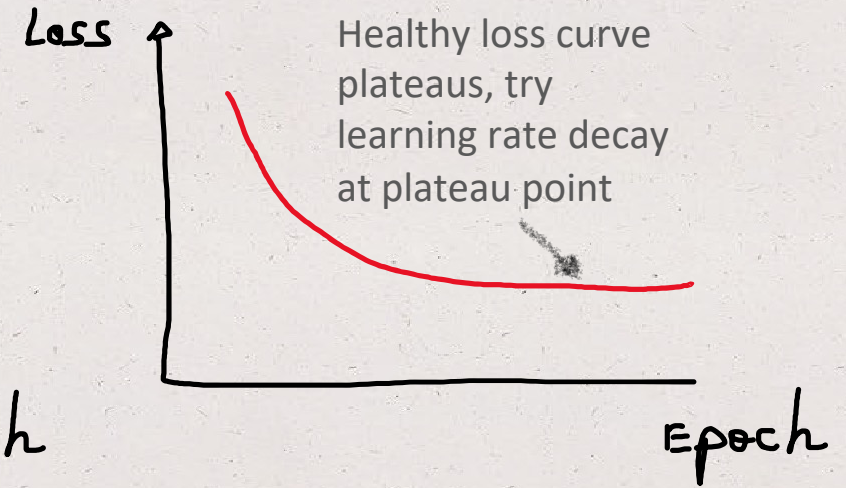
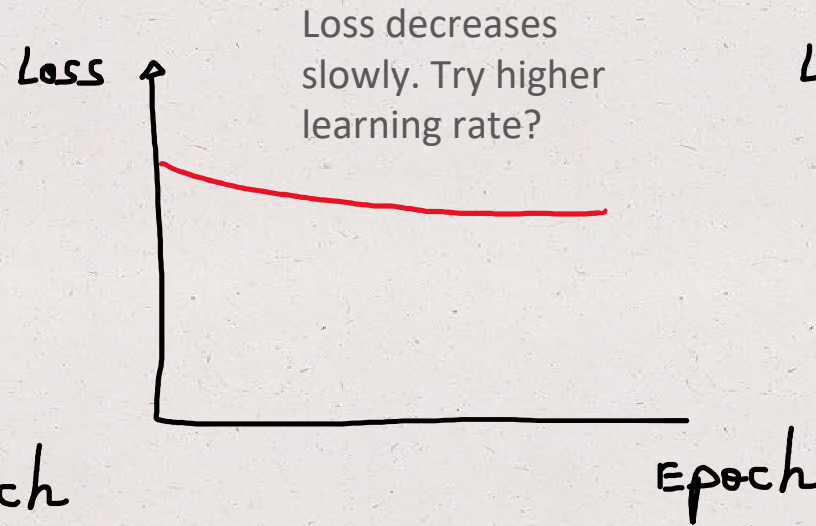
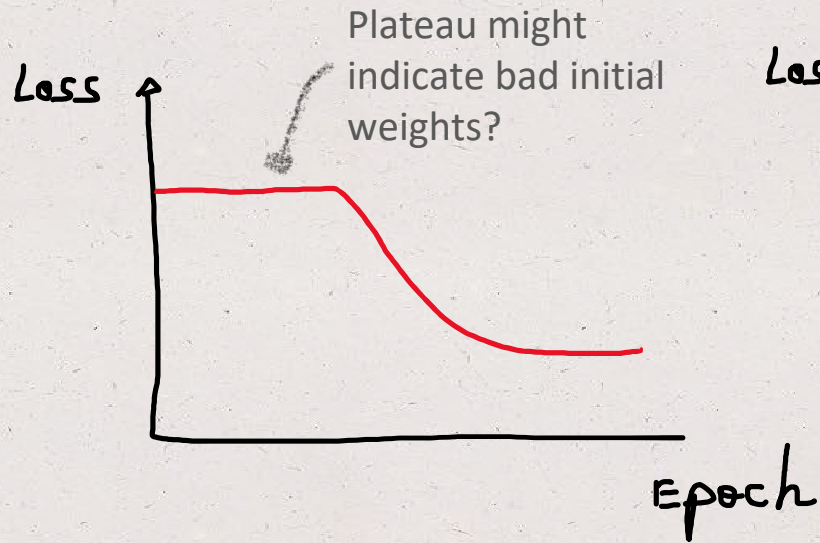


# Debugging learning curves





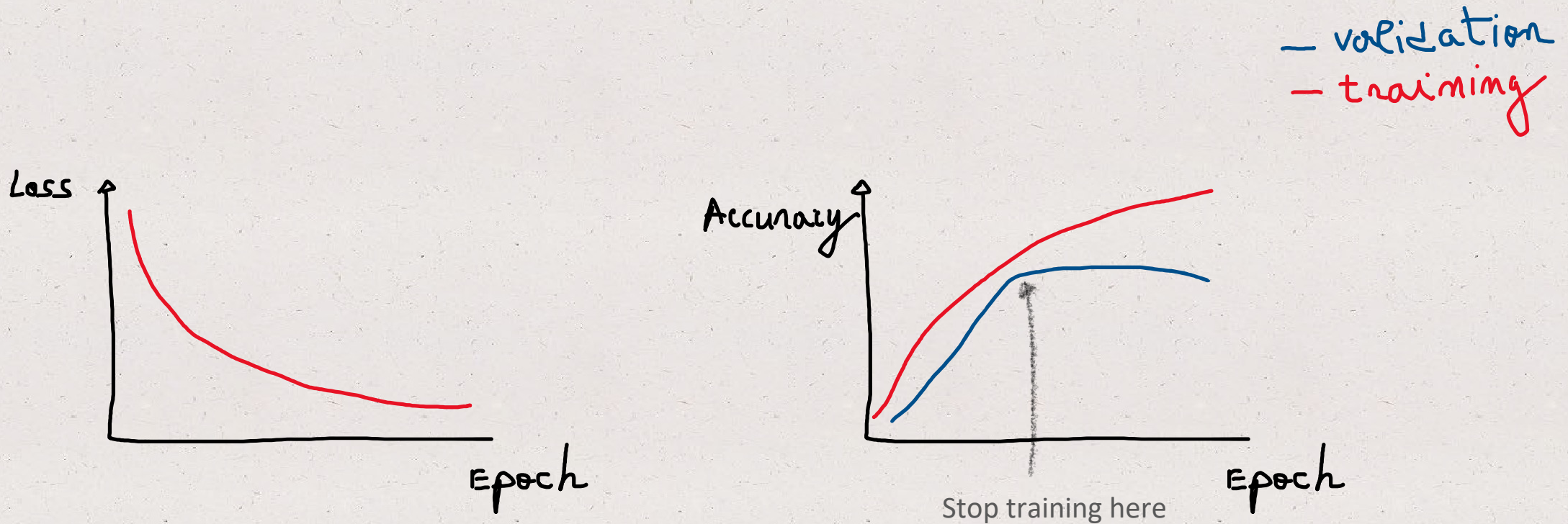
# Debugging learning curves



— validation  
— training



# Debugging learning curves



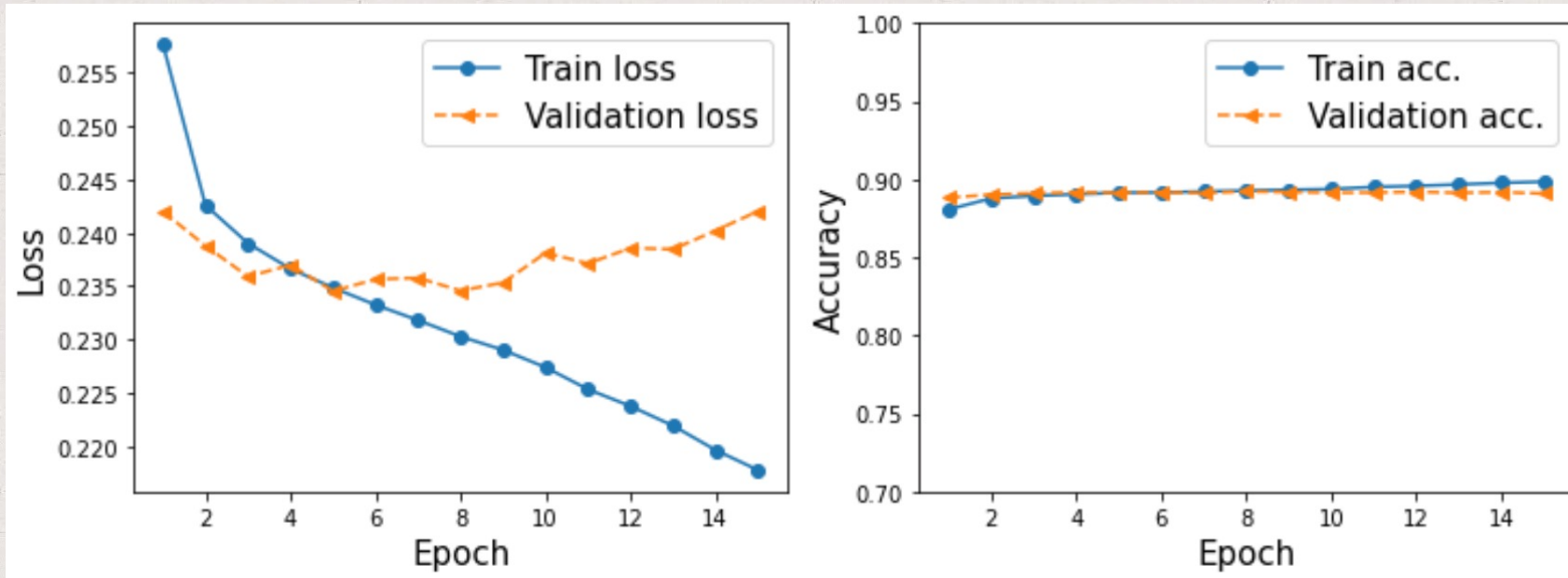
Early stopping:

stop training when accuracy on  
validation decreases.



# Debugging learning curves

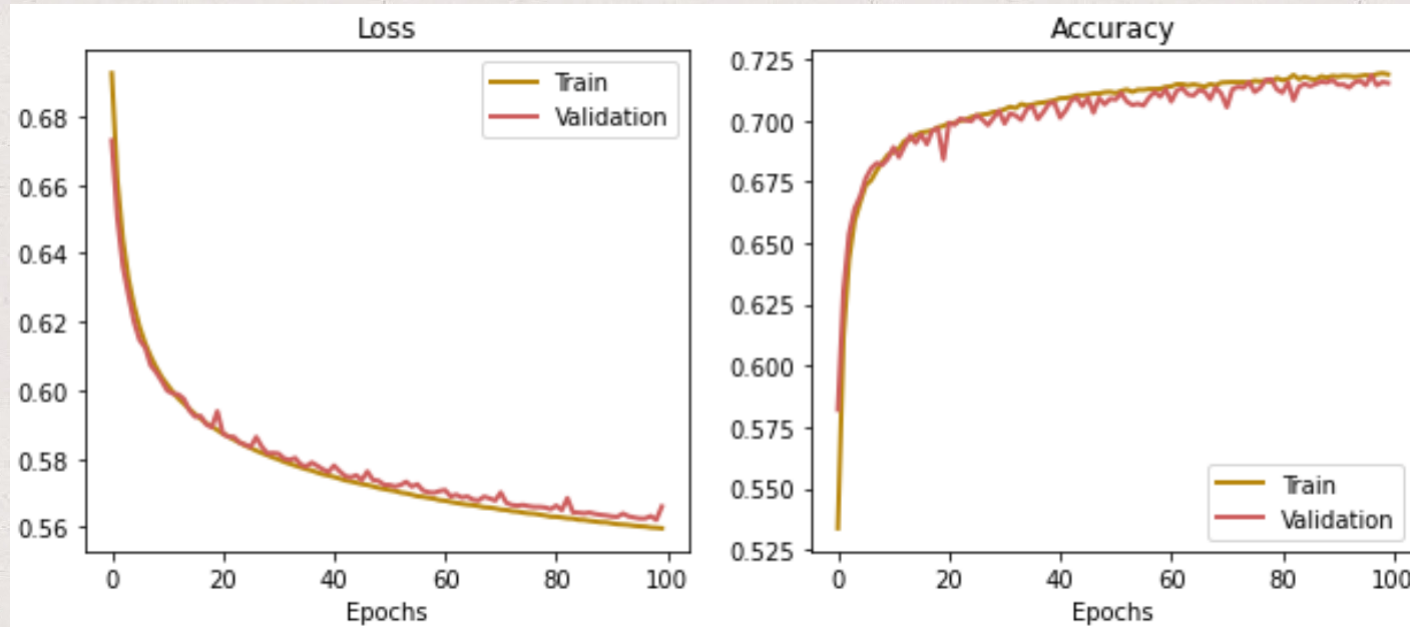
What do you recommend?





# Debugging learning curves

What do you recommend?





# My research: RNN/LSTM vs. Linear model

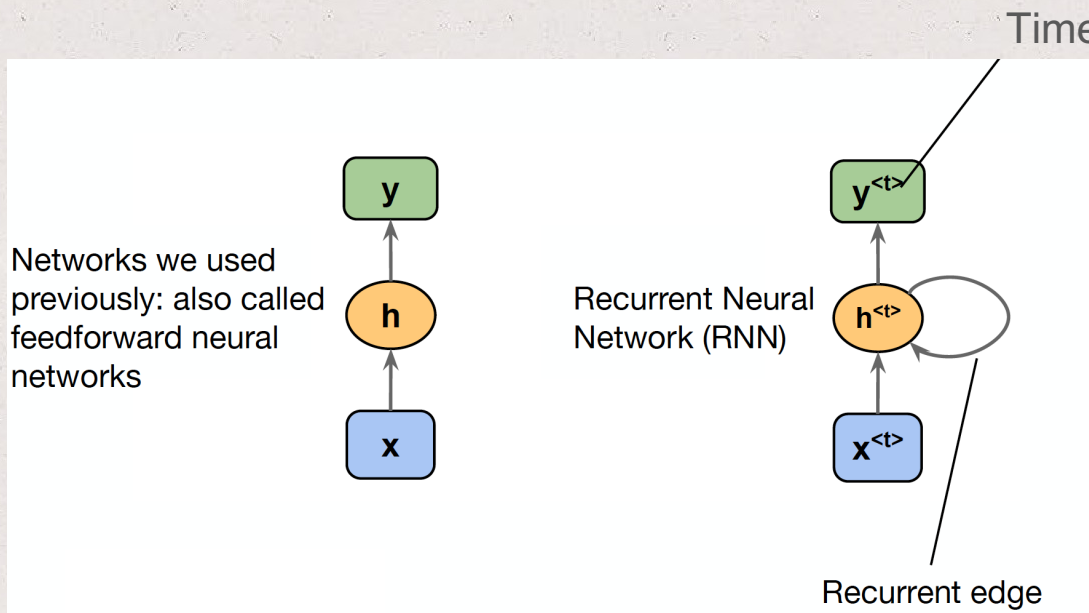
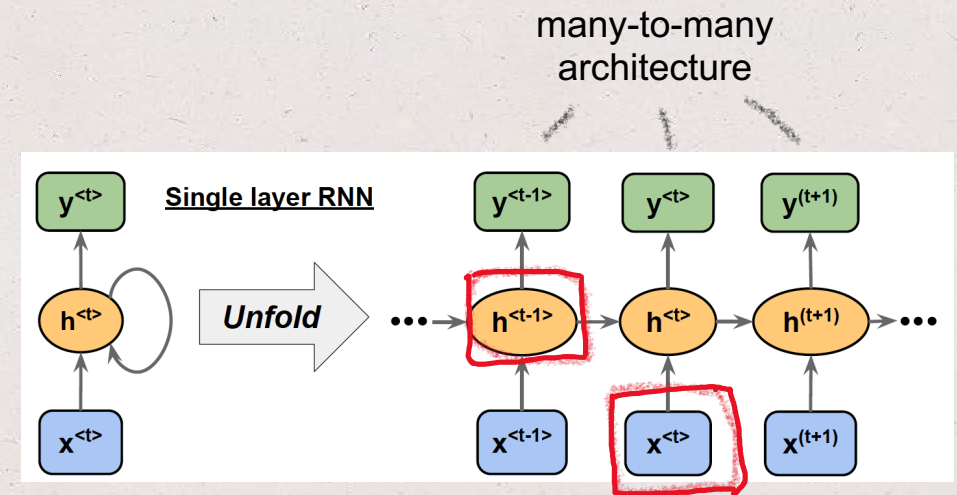


Image source: Sebastian Raschka, Vahid Mirjalili. *Python Machine Learning, 3rd Edition*. Packt, 2019



Each hidden unit receives two inputs

**Data:** Electronic medical records (EMR)

**Question:** Predict diagnosis in the next hospital visit



# My research: RNN/LSTM vs. Linear model

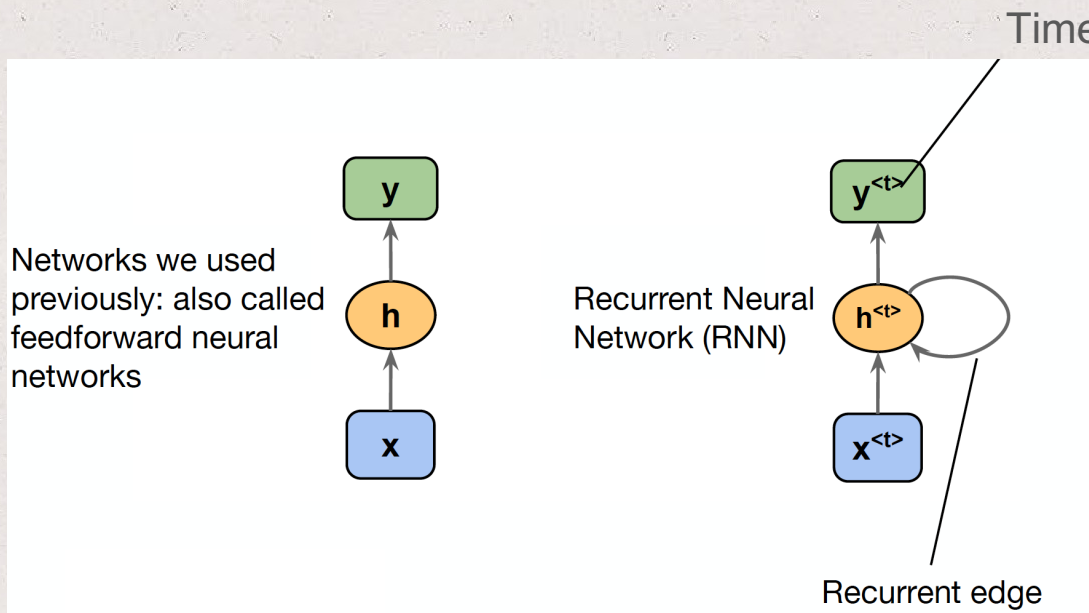
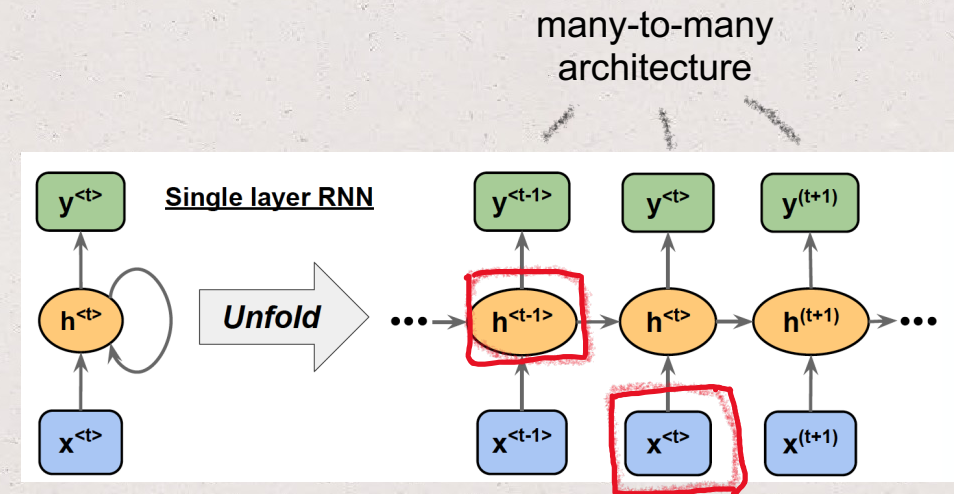


Image source: Sebastian Raschka, Vahid Mirjalili. *Python Machine Learning, 3rd Edition*. Packt, 2019



**Data:** Electronic medical records (EMR)

**Question:** Predict diagnosis in the next hospital visit

**Similar data:** MIMIC II (<https://archive.physionet.org/mimic2/>)