STAT430: Machine Learning for Financial Data

LarryHua.com/teaching

Spring 2019

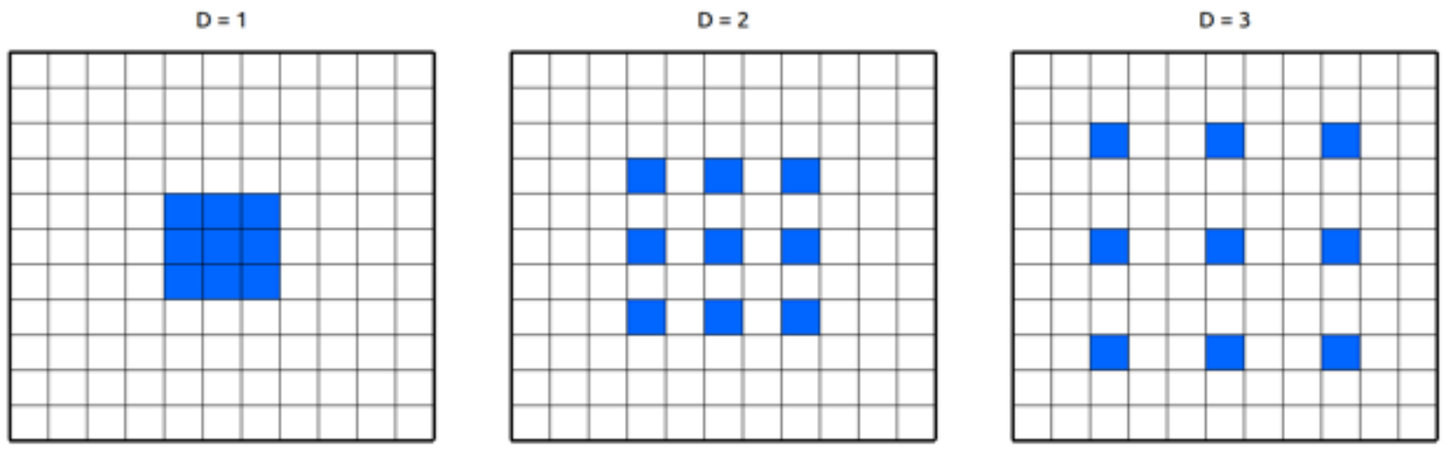
1D CNN for sequence data

1D convnet

- similar to 2D and 3D convnets, but the convolution window is a 1D window on the temporal axis
- In Keras, `layer_conv_1d()`
  - input: `(samples, time, features)`
  - output: similarly shaped 3D tensors
  - recall that the input for `layer_conv_2d()` is `(samples, height, width, features)`
- unlike 2D or 3D convnets, 1D convnets can have larger convolution windows, say sizes 7 or 9
- 1D convnet is a fast and cheap alternative to a recurrent network, especially for tasks such as sentiment analysis where the order of the positive/negative words in the sequence is not very relevant
  - eg, useful for movie review classification, but not very useful for predicting the temparature in an hour

Combine CNN and RNN

- one could stack many convolution and pooling layers to recognize longer-term patterns, but still a weak method for order-sensitive tasks
- better way: CNN + RNN
  - use a 1D convnet as a preprocessing step before a RNN
    - the convnet turns the long input sequence into much shorter ones of higher-level features
    - the shorter sequence of extracted features then becomes the input to the RNN
  - especially useful for long sequences that cannot be processed with RNNs, such as sequences with thousands of steps
- a dilated kernel used in CNN might be useful



Generator functions for data input

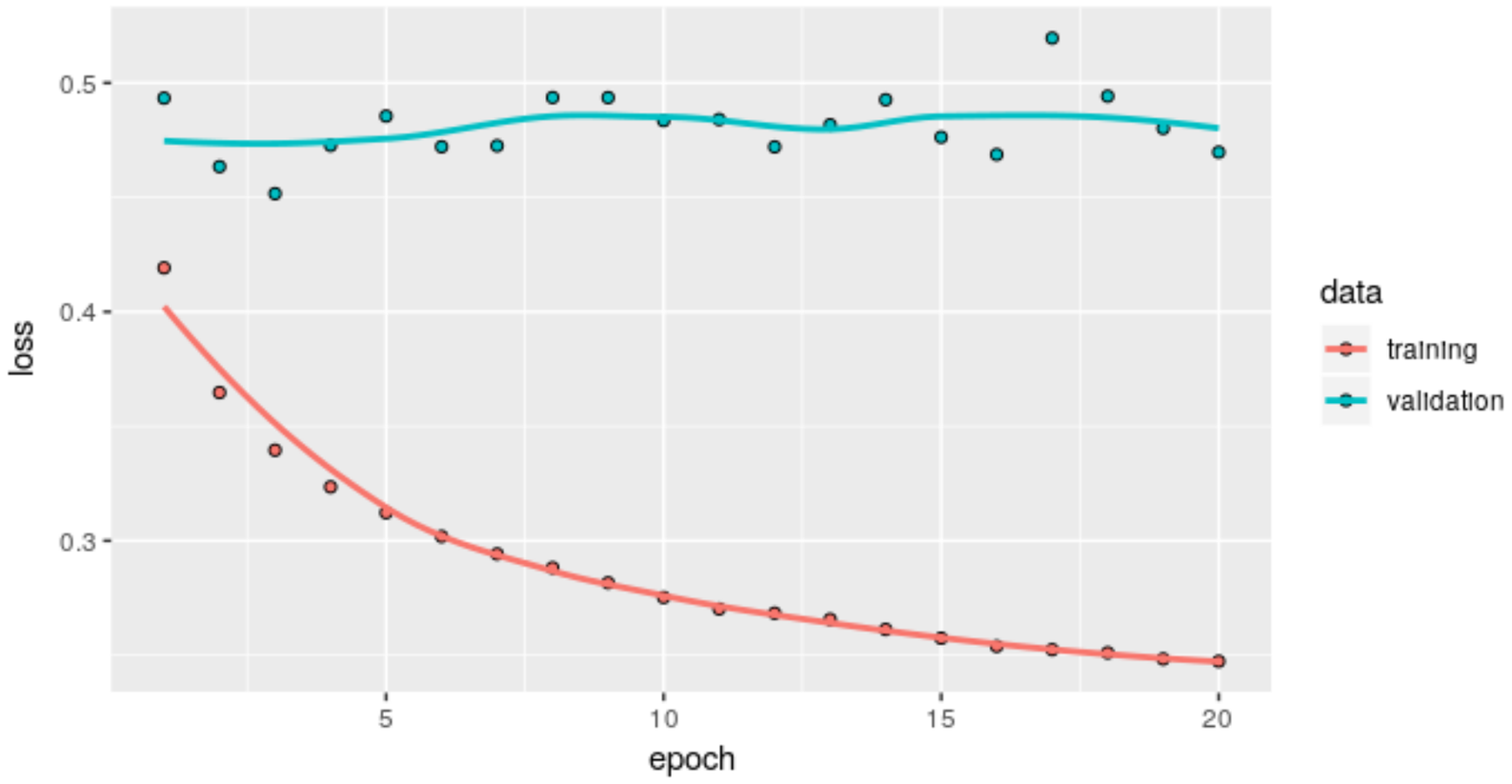
- a generator function is a special type of function that is called repeatedly to obtain a sequence of values
  - when data is large
  - when a particular way of generating batches is required
- 2 common ways to sample batches
  - randomly sample from the set
    - eg, `rows <- sample(1:nrow(data), batch_size)`
  - sample batches in a particular way, eg, in a sequential order. Then we need a variable to store the status of each batch
    - use superassignment: **<<-**

```
generator <- function(i0){i <- i0; function() {i <<- i + 1; i}}
gen <- generator(10)
gen(); gen()
```

```
## [1] 11
```

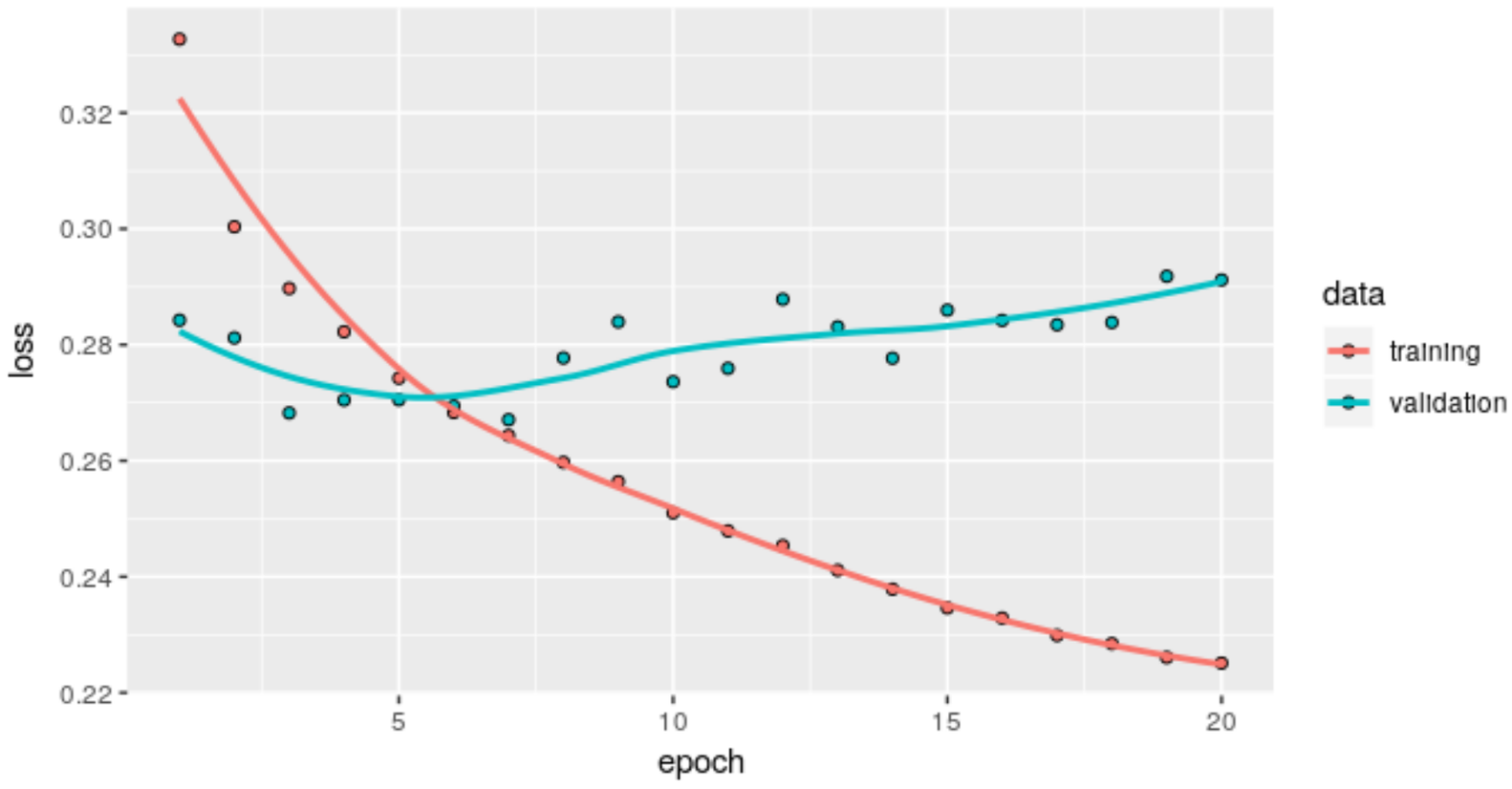
```
## [1] 12
```

Examples - 1D CNN for temperature

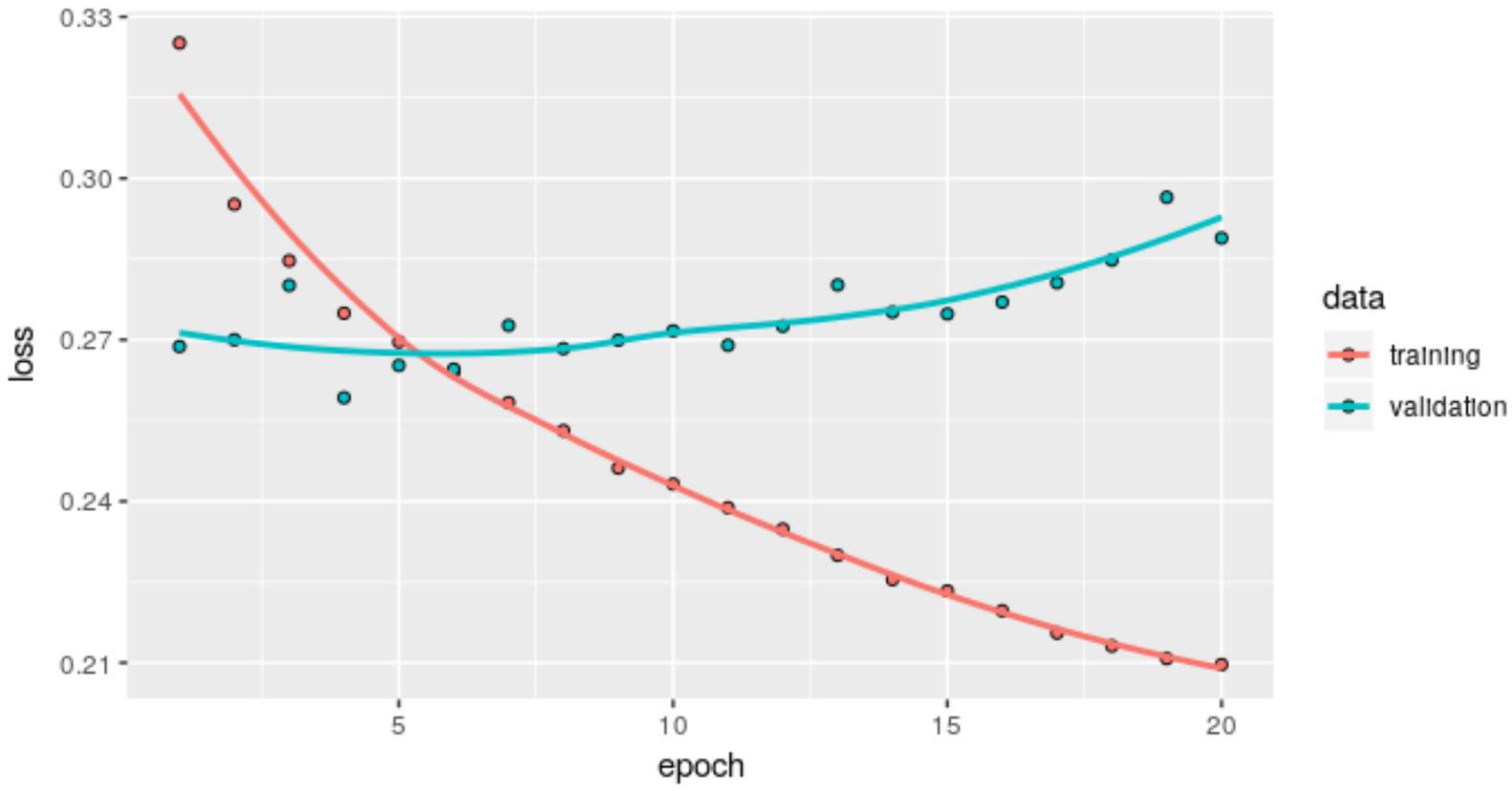


- Try R

Examples - 1D CNN + RNN for temperature



Examples - 1D CNN + RNN for temperature



* 1D CNN + dilated kernel

Examples - LOB analysis

- Comparisons of different architectures
  - Model A: 2D CNN (feature dimension = 1)
  - Model B: RNN (feature dimension = 20)
  - Model C: 1D CNN + RNN (feature dimension = 20)
- Try R

Takeaways for 1D CNN

- 1D convnets perform well for processing temporal patterns, and they are faster alternative to RNNs on some problems
- 1D convnets are structured much like their 2D equivalents of images and consist of stacks of `layer_conv_1d()` and `layer_max_pooling_1d()`
- Because RNNs are extremely expensive for processing very long sequences, we can use a 1D convnet as a preprocessing step before a RNN, shortening the sequence and extracting useful representations for the RNN to process
- Back to Course Scheduler