STAT430: Machine Learning for Financial Data

LarryHua.com/teaching

Spring 2019

Fundamentals and workflows

Four branches of machine learning

- **Supervised learning**
  - Classification
  - Regression
  - Sequence generation: Given a picture, predict a caption describing it
  - Syntax tree prediction: Given a sentence, predict its decomposition into a syntax tree
  - Object detection: Given a picture, draw a bounding box around certain objects inside the picture
  - Image segmentation: Given a picture, draw a pixel-level mask on a specific object
- **Unsupervised learning**
  - Dimensionality reduction and clustering
  - Often a necessary step in better understanding a dataset before supervised-learning
- **Self-supervised learning**: supervised learning w/o human-annotated label, and labels are generated from data via some algorithms
- **Reinforcement learning**: an agent receives information about its environment and learns to choose actions that will maximize some reward

Data preprocessing for neural networks

- Vectorization: All inputs and targets in a neural network must be tensors of floating-point data
- Value normalization
  - Take small value: most values should be in the 0–1 range
  - Be homogeneous: all features should take values in roughly the same range
  - R: `scale()`
- Missing values
  - Assign 0 for missing values if 0 is not a meaningful value
  - If missing in test set, but not in training set:
    - Copy some training samples several times, and drop some features from the training samples that are likely to be missing in the test set

Feature engineering in the era of deep learning

- Not as important as that for shallow learning, but can be beneficial:
  - Good features allow using fewer resources to solve problems more elegantly
  - Good features allow solving a problem with far less data

Overfitting and regularization

- Overfitting is always an issue in machine learning
  - The best solution is to get more training data
  - The 2nd best solution is **regularization**
- Regularization
  - Reduce learnable parameters / capacities
  - Weight regularization: adding to the loss function of the network a cost associated with weights
    - L1 regularization: proportional to the absolute value of the weights
      - `layer_dense(..., kernel_regularizer = regularizer_l1(0.001), ... )`
    - L2 regularization: proportional to the square of the value of the weights
      - `layer_dense(..., kernel_regularizer = regularizer_l2(0.001), ... )`
    - `layer_dense(..., kernel_regularizer = regularizer_l1_l2(l1 = 0.001, l2 = 0.001), ... )`
    - The cost is only added at training time, so the loss for this network will be much higher at training time than at test time
- Dropout

Dropout

- Applied to a layer, randomly setting to zero a number of output features of the layer during training
  - eg: for a given layer, the output is [0.2, 0.5, 1.3, 0.8, 1.1], and after dropout it becomes [0, 0.5, 1.3, 0, 1.1]
- Dropout rate: the fraction of the features that are zeroed out: (0.2 ~ 0.5)
- Apply `layer_dropout(rate = 0.5)` immediately before the layer
- Try R

Workflow of machine learning

- Defining the problem and assembling a dataset
- Choosing a measure of success
- Deciding on an evaluation protocol
    - Maintaining a hold-out validation set: when plenty of data is avaliable
    - Doing K-fold cross-validation: when too few samples for hold-out validation
    - Doing iterated K-fold validation: when little data is available
- Preparing data
- Developing a model that does better than a baseline
    - two preliminary hypothesis:
        - outputs can be predicted by inputs
        - the available data is sufficiently informative to learn the relationships between inputs and outputs
    - Last-layer activation / Loss function / optimizers
- Developing a model that overfits: add more and bigger layers, train for more epoches
- Regularizing your model and tuning your hyperparameters

Last-layer activation and loss function

- Binary classification
  - sigmoid | binary_crossentropy
- Multiclass, single-label classification
  - softmax | categorical_crossentropy
- Multiclass, multilabel classification (non-exclusive labels for each observation)
  - sigmoid | binary_crossentropy
- Regression to arbitrary values
  - none | mse
- Regression to values between 0 and 1
  - sigmoid | mse or binary_crossentropy
- Back to Course Scheduler