Teaching a Computer to Fish

Jack Burdick

Last Update: April 12, 2018

Contents

Part I

Background

Chapter 1

Introduction

But if you teach your computer to fish..

There are many great resources that exist.

I wanted to create the guide I wish I found when I started learning tensorflow.

Chapter 2

Resources and Communities

There are many great resources and communities that I'd like to highlight

Online Communities

- Reddit
- Stack Overflow
- Slack

Blogs

- XXXXXXXXXXXXXXXX

Online Courses

- Udacity
- XXXXXXXXXXXXXXXX

Text Books

- XXXXXXXXXXXXXXXX

Chapter 3

Brief Walkthrough

Blah....

Environment

Overview of Environment

Terminal

Hardware

CPU vs GPU

Cloud Providers

AWS Quickstart

Python

This section will not teach how to program in python. Rather, common functionality as well as common XXXXX areas will be introduced.

Datatypes

Tuple

fixed-length immutable sequence of Python objects.

List

variable-length mutable sequence of Python objects.

Dict

Set

Functions

Built-in Sequence Functions

enumerate

sorted

zip

reversed

Generators

Errors and Exception Handling

IO

Other

```
######################
# Sample code snippet
######################
import numpy as np

def incmatrix(genl1,genl2):
   m = len(genl1)
   n = len(genl2)
   M = None #to become the incidence matrix
   VT = np.zeros((n*m,1), int) #dummy variable

   #compute the bitwise xor matrix
```

```
M1 = bitxormatrix(genl1)
M2 = np.triu(bitxormatrix(genl2),1)

for i in range(m-1):
   for j in range(i+1, m):
      [r,c] = np.where(M2 == M1[i,j])
      for k in range(len(r)):
         VT[(i)*n + r[k]] = 1;
         VT[(i)*n + c[k]] = 1;
         VT[(j)*n + r[k]] = 1;
         VT[(j)*n + c[k]] = 1;

         if M is None:
            M = np.copy(VT)
         else:
            M = np.concatenate((M, VT), 1)

         VT = np.zeros((n*m,1), int)

return M
```

git

Overview

Commands

Github

Jupyter

Environment

Styling

Reloading Module Dependencies

Profiling

Anaconda

Docker

Common Libraries

Overview of Environment

Numpy

Designed to work with homogeneous numerical array data

ndarrays

Initialization

Indexing

Datatypes

Arithmetic

Basic

Statistical Methods

IO

Other

transpose

Set Logic

Images

OpenCV

Natural Language Processing

NLTK

Ingesting Data

scrapy

beautifulsoup

sql

mongo

Analyzing Data

Pandas

Designed to work with tabular or heterogeneous data

Series

Dataframe

Hierarchical Indexing

Describing and Visualizing

Merging, Joining, Pivoting

Groups

Data Loading

Visualizing Data

Matplotlib

Basics

Representation of types of data

Categorical Variables

Frequency Distribution Tables two columns, one for the category and the other for the number of occurrences (frequency)

Bar Charts Shows a table in a graphical form where each bar (each different category) height/length is representative of the value

Pie Charts Shows a table in a graphical form where a circle (pie) shows the relative frequency of each categorical value

Pareto Diagrams a special type of bar chart where the categories are shown in descending order of frequency and an additional curve shows the cumulative frequency (sum of relative frequencies)

Numerical Variables

Figures, Subfigures

Chart Type Examples

Line

Scatter

Bar

Histograms

Pie

Customization

Colors

Markers

Ticks

Labels

Legends

Annotations

Saving to File

Predicting Data

Scikit-Learn

Transformation Pipelines

Training

Cross-Validation

Fine-Tuning

Hyper-Parameter Optimization

Grid Search

Randomized Search Disucessed in ref to other setion

Data Provanece and Reproducibility

Pachyderm

Others

Regular Expressions

brief overview and examples

tangent

Markdown

Part II

ML

Chapter 4

Basics

categorical or numerical. Numerical can be discrete or continuous

qualitative or quantitative.

Qualitative can be nominal (aren't numbers and can't be put in any order – e.g. the seasons: spring, summer, fall, winter) or ordinal (groups and categories that follow a strict order – e.g. difficult levels: hard, medium, or easy)

Quantitative are represented by numbers but can be interval (0 is meaningless – e.g. temperature in C or F, where true zero is not 0) or ratio (has a true 0 – e.g. temperature in K, weight or length)

Acquiring Data

Resources

Data Pre-processing

Handling Missing Data

Filtering Out

Filling In

Handling Categorical Data

Encoding

Feature Scaling, Normalization

Min-Max scaling (Normalization)

values are shifted and rescaled so they end up on a [0,1] range

Standardization

first, subtracts mean, then divides by variance

Others

Removing Duplicates

Outliers

Discretization and Binning

Partitioning Data

Sampling

Training, validation, test

Some Terms

input variable(s) – predictors, independent variables, features, or simply variables.

output variable(s) – response or dependent variable

relationship $Y = f(x) + \epsilon$ estimate f. prediction and inference.

reducible error – the estimated function $\hat{f}$ will likely not be perfect, and the reducible error is the error that could be corrected. The *irreducible error* is an error that can not be corrected. The irreducible error may be larger than zero due to *unmeasured variables e.g.* varibles that were not measured and *unmeasurable variation e.g.* an individual's feelings/emotions or variation in the production of a product. The irreducible error provides an upper bound on the performance of the predicted $\hat{f}$

Supervised vs Unsupervised

supervised

–

unsupervised

– observe input variables without corresponding output values.

Classification vs Regression

Regression

– predicting a continuous or quantitative output value

Classification

– predicting categorical or qualitative output value (such as a non-numerical value)

Bayes Classifier

Training

Quality of Fit

Regression Example

Mean Squared Error. $\hat{f}(x_i)$ is the prediction that $\hat{f}$ produces for the ith sample. The output will be small for predicted values that are similar to the ground truth

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2 \tag{4.1}$$

Classification Example

The proportion of mistakes that are made.

$$error\_rate = \frac{1}{n}\sum_{i=1}^{n}(y_i \neq \hat{y}_i) \tag{4.2}$$

$\hat{y}_i$ is the predicted classification label for the ith observation using our predictor/model $\hat{f}$ and y_i is the ground truth label

(Over—Under)fitting

Overfitting

Overfitting refers to a case in which a model fits the training data very well but does not fit validation/test set

Underfitting

Bias Variance Trade-off

Variance

variance refers to the amount the model would change if it was trained/estimated using a different training data set

Bias

Bias refers to the amount of error that is introduced by approximating a problem with a model that is simpler than the complex problem For example, linear regression assumes a linear relationship between the features and labels. However, it is unlikely that a true linear relationship exists and so using linear regression to model this type of particular problem will likely introduce some bias.

Trade-Off

In general, as a more "flexible" model is used, the variance will increase and the bias will decrease.

It is easy to obtain a model with low bias but high variance (*e.g.* drawing a squiggly line through every training observation) and it is easy to obtain a model with low variance but high bias (*e.g.* drawing a straight line approximating every training observation) but it is difficult to obtain a model that has both low variance and low bias.

It should be noted that in a real world example, it maynot be possible to explicitly calculate the test error, bias, or variance.

Parametric vs non-parametric

Cost is frequently used interchangeably with loss. Technically, loss refers to the error on a single example and cost is the average of the loss across the entire training set.

One-versus-all *OvA* (also *one-versus-rest*)

One-versus-one (OvO) – train a binary classifier for every pair

Pred		Ground Truth: Positive	Ground Truth: Negative
	Positive	TP	FP
	Negative	FN	TN

Table 4.1: Example confusion matrix

Metrics

- *Accuracy*, (Eq. 4.3): the ratio of correct predictions to the total number of predictions.

$$\frac{TP+TN}{TP+TN+FP+FN} \tag{4.3}$$

- *Sensitivity*, (Eq. 4.4): the ratio of true positives that are correctly identified.

$$\frac{TP}{TP+FN} \tag{4.4}$$

- *Precision*, (Eq. 4.5): the ratio of positives that are, in fact, positive. If the classifier predicts positive, how often is is correct?

$$\frac{TP}{TP+FP} \tag{4.5}$$

- *AUC (Area Under the Curve)*, is a single value representing the area under an ROC curve. Though generally referred to as the AUC, the term is correctly abbreviated AUROC, specifying that the curve is an ROC curve.

Chapter 5

Foundational Methods

Regression

Simple Linear Regression

$$Y \approx \beta_0 + \beta_1 X \tag{5.1}$$

$\approx$ can be read as "*is approximately modeled as*". Y is a quantitative response (output/prediction) and X predictor variable(input/feature). β_0 and β_1 are two unknown constants representing the intercept and slope, respectively. These unknown values that determine the behavior of the model are known as the model *parameters* or *coefficients*

Multiple Linear Regression

Using n predictors:

$$Y \approx \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n \tag{5.2}$$

Polynomial Regression

K-Nearest Neighbors

The optimal value for k will depend on the bias-variance trade-off

Classification

Logistic Regression

Chapter 6

Term dump

Collinearity – When two or more predictor variables are closely related to one another they are said to be collinear.

Curse of Dimensionality –

dummy variable –

Population vs Sample – the population (usually denoted N) is the collection of all the items of interest in a study where as the sample is a subset of a population (usually denoted n). The numbers obtained when working with a population are called the 'parameters' and the numbers obtained when working with a sample are a called 'statistics'. a random sample is obtained when each member of the sample is chosen from the population by chance and accurately reflects the population

Chapter 7

ML with TensorFlow

Overview

Introduction

Artificial Neural Networks

Convolutional Neural Networks

Recurrent Neural Networks

Weight Initialization

Activation Functions

Optimizers

What are optimizers

Types of optimizers

TF optimizers

tf.train.GradientDescentOptimizer

tf.train.MomentumOptimizer
tf.train.RMSPropOptimizer
tf.train.AdadeltaOptimizer
tf.train.AdagradOptimizer
tf.train.AdagradDAOptimizer
tf.train.AdamOptimizer
tf.train.FtrlOptimizer
tf.train.ProximalGradientDescentOptimizer
tf.train.ProximalAdagradOptimizer

Advanced

Gradient Computation

Gradient Clipping

Hyperparameters

Training Related

Learning Rate

Batch Size

Number of Training Iterations

Momentum

Weight Update

SGD, CG, L-BFGS, more complex more hyper-parameters

Stopping Criteria

Model Related

Architecture

Weight Initialization

Weight-decay

L1

L2

Drop-out

Hyper-parameter optimization

OVERVIEW

Coordinate Descent

All hyper-parameters remain fixed, except for the hyper-parameter of interest. The hyper-parameter of interest is then adjusted such that the validation error is minimized.

Grid Search

Random Search

Automated / Model-based Methods

Regulariazation

Image Augmentation

Serving

TensorBoard

Estimators

Metrics

Eager

Model Persistence

Part III

End To End Examples

Chapter 8

EndToEnd

Structured

Linear Regression

KNN

Image

Image Classification

Image Segmentation

Adversarial Exmaples

AutoEncoder

Generative Adversarial Network

TimeSeries

Text

Sentiment Analysis

Audio

Audio to Text

Part IV

Moving Forward

Conclusion

Part V

Dump Space - may/maynot include

Chapter 9

IOT

Micro Processor

Raspberry Pi

Micro Controller

Arduino

Chapter 10

Frequent Roadblocks

Excel

CSV

Chapter 11

Ubuntu

Trouble Shooting

I/O

bluetooth

Audio

https://askubuntu.com/questions/833322/pair-bose-quietcomfort-35-with-ubuntu-over-bluetooth

Part VI

Research