



Golang Machine Learning Libraries

Machine Learning Golang libraries



If there's one thing that everyone (yes, including non-CS people) have hyped incessantly over the past few years, it's Machine Learning. And while we know Go provides faster speeds overall at accessing requests over the internet, it would be really disappointing if it couldn't be used for something practical – like Machine Learning. Well, this article is designed to give you a brief introduction into the applications of Golang in ML about what is possible currently, and what is still in the works.

Machine Learning

As a brief overview of ML if you haven't heard of it already, the best definition of ML by Tom Mitchell has been accepted:

*A computer program is said to **learn** from experience E with respect to some class of tasks T and performance measure P , if its performance at*

tasks in T, as measured by P, improves with experience E.

– Machine Learning 1983-86, Tom Mitchell

ML is a method of teaching the computer to learn things in a manner of humans – by trial, error and updation of it's "understanding" of the solution.

The work behind Machine Learning is nothing new – it started when Andrey Markov tried analysing the contents of a poem, resulting in his discovery of Markov Chains. It has led scientists to create better algorithms throughout the 1970s and 1980s, and it had all culminated to 2016, when a machine named AlphaGo beat professional players in the game of Go (not related to Golang).

Name	Score
AlphaGo Fan	5 : 0 against Fan Hui
AlphaGo Lee	4 : 1 against Lee Sodol
AlphaGo Master	60 : 0 against professional players online
AlphaGo Zero	100 : 0 against AlphaGo Lee 89 : 11 against AlphaGo Master

Scores of various AlphaGo models against real players

So, Machine Learning works ! And Python, being one of the top 3 programming languages for the past decade, does it quite well. So the question is, can Go match up? I think the answer might surprise you.

Gorgonia and Goro

First, let's talk about [Gorgonia](#), the closest that GoLang has gotten to Keras and TensorFlow in Python. Keras is a Python ML/Data Science library that is very intuitive in it's creation of Neural Networks, the fundamental building blocks to Data Science and deep pattern recognition.

Gorgonia is a huge project, that has accomplished some big feats:

Can perform automatic differentiation

Can perform symbolic differentiation

Can perform gradient descent optimizations

Can perform numerical stabilization

Provides a number of convenience functions to help create neural networks

Is fairly quick (comparable to Theano and Tensorflow's speed)

Supports CUDA/GPGPU computation (OpenCL not yet supported)

Will support distributed computing

The backend is a little difficult to understand, especially for beginners since it uses a "Computation Graph", which simply means that the program creates a hierarchical tree structure as the source code is read, making it easier for machines to parse through the program instead of trying to interpret the order of variable and function calls.

The library is easy to pick up and use. Install the package with

```
1 import ("gorgonia.org/gorgonia")
```

and you're good to go! It also has a handy "examples" subdirectory with many popular datasets like MNIST pre-loaded.

```
1 inputs, targets, err := mnist.Load(typ, "./testdata", tensor.Float64)
```

This would be a good time to mention a separate project based on Gorgonia called Goro. In Goro, it is possible to create a Convolutional Neural Network (CNN) sequentially:

```
1 import (
```

```
2     . "github.com/aunum/goro/pkg/v1/model"
3     "github.com/aunum/goro/pkg/v1/layer"
4 )
5
6 //Define x_train and y_train
7
8 model, _ := NewSequential(model_name)
9
10 model.AddLayers(
11     layer.Conv2D{Input: 1, Output: 32, Width: 3, Height: 3},
12     layer.MaxPooling2D{},
13     layer.Conv2D{Input: 32, Output: 64, Width: 3, Height: 3},
14     layer.MaxPooling2D{},
15     layer.Conv2D{Input: 64, Output: 128, Width: 3, Height: 3},
16     layer.MaxPooling2D{},
17     layer.Flatten{},
18     layer.FC{Input: 128 * 3 * 3, Output: 100},
19     layer.FC{Input: 100, Output: 10, Activation: layer.Softmax},
20 )
21
22 optimizer := g.NewRMSPropSolver()
23
24 model.Compile(xi, yi,
25     WithOptimizer(optimizer),
26     WithLoss(m.CrossEntropy),
27     WithBatchSize(100),
28 )
29
30 model.Fit(x_train, y_train)
```

It's very intuitive, and anyone who's used keras previously can work on it directly.

MLGo, GoML and GoLearn

Both [mlgo](#) and [goml](#) aim to extend the capabilities and functions of the current ML libraries by implementing algorithms that are required in the process.

While [mlgo](#) seems to have mainly focused on developing stronger algorithms for Clustering, [goml](#) has implemented some handy Regression models – Ordinary Least Squares, Linear Regression, Softmax Regression, Batch

Gradient Descent, etc. which are required in the CNN layers for approaching the solution. **goml** has successfully implemented a few naive bayes classification methods, which are still not comparable to what can be achieved in Python... But it's a start !

mlgo is also working on several density based and hierarchical clustering methods which are essential in unsupervised learning models.

Golearn, on the other hand, is taking a simplistic approach to building built-in functions that can be referenced as instances directly in a .go code file to train and predict on machine learning datasets. These are restrictive in their number of parameters and layers, but perhaps easier for a beginner to understand. To show how easy it is to work with **golearn**, look at the code below.

```
1  package main
2
3  import (
4      "fmt"
5
6      "github.com/sjwhitworth/golearn/base"
7      "github.com/sjwhitworth/golearn/evaluation"
8      "github.com/sjwhitworth/golearn/knn"
9  )
10
11 func main() {
12     // Load in a dataset, with headers. Header attributes will be stored
13     // Think of instances as a Data Frame structure in R or Pandas.
14     // You can also create instances from scratch.
15     rawData, err := base.ParseCSVToInstances("datasets/iris.csv", false)
16     if err != nil {
17         panic(err)
18     }
19
20     // Print a pleasant summary of your data.
21     fmt.Println(rawData)
22
23     //Initialises a new KNN classifier
24     cls := knn.NewKnnClassifier("euclidean", "linear", 2)
25
26     //Do a training-test split
27     trainData, testData := base.InstancesTrainTestSplit(rawData, 0.50)
```

```
28     cls.Fit(trainData)
29
30     //Calculates the Euclidean distance and returns the most popular labels
31     predictions, err := cls.Predict(testData)
32     if err != nil {
33         panic(err)
34     }
35
36     // Prints precision/recall metrics
37     confusionMat, err := evaluation.GetConfusionMatrix(testData, predictions)
38     if err != nil {
39         panic(fmt.Sprintf("Unable to get confusion matrix: %s", err.Error()))
40     }
41     fmt.Println(evaluation.GetSummary(confusionMat))
42 }
```

There are many other interesting libraries by developers around the world, trying to build up a community for Go and making it user intuitive while maintaining the innate vision of scale that Golang is built for. It is also possible to join these communities and be a contributor to these open source libraries.

So Go on !

A great discussion on the future of ML in Golang:

https://www.reddit.com/r/golang/comments/anbjin/is_there_a_future_in_machine_learning_in_go/

References

<https://code.google.com/archive/p/mlgo/>

<https://gorgonia.org/>

<https://github.com/cdipaolo/goml>

[← Previous Post](#)

[Next Post →](#)

Leave a Comment

Your email address will not be published. Required fields are marked *

Type here..


Name*

Email*

Website

Save my name, email, and website in this browser for the next time I comment.

Post Comment »

Search ... 

Recent Posts

[Remote Debugging in Golang and Java](#)

[GoLang Read File Line by Line](#)

[GoLang Global Variables](#)

[Techniques to Maximize Your Go Application's Performance](#)

[Guide to Cloning Git Repositories](#)

[Content Delivery Network: What You Need to Know](#)

[7 Most Popular Programming Languages in 2021](#)

[GoLang Byte Array to String](#)

[Golang int64 to string conversion](#)

[What is rune in GoLang?](#)

Favorite Sites

[AskPython](#)

[CodeForGeek](#)

[JournalDev](#)

[LinuxForDevices](#)

Copyright © 2022 · GoLang Docs · GoLangDocs is part of JournalDev IT Services Private Limited