Comia, Mico Ellerich M.
May 22, 2021

I . BlazingText

Summary:

Blazing Text is a built-in Amazon SageMaker algorithm that provides “highly optimized implementations of the Word2vec and text classification algorithms.” It allows you to scale to large datasets easily through utilization of multi-core CPU or GPU setups. It is mainly used for natural language processing tasks or text classification.

Practical Problems it can solve:

1. Sentiment Analysis
2. Machine Translation

Terms and Definitions:

- **Word embedding** - the resulting high-quality distributed vector representation of a word.
- **Skip-gram** - an unsupervised learning technique used to find the most related words for a given word.
- **Continuous bag-of-words (CBOW)** - a continuous representation of text as the bag of its words, disregarding grammar and word order but keeping multiplicity.

Notes: Training

- **Train with File Mode**

__label__4 linux ready for prime time , intel says , despite all the linux hype , the open-source movement has yet to make a huge splash in the desktop market . that may be about to change , thanks to chipmaking giant intel corp .

```
__label__2  bowled by the slower one again , kolkata , november 14 the past caught up with sourav ganguly as the indian skippers return to international cricket was short lived .
```

- **Train with Augmented Manifest Text Format**

```
{"source":"linux ready for prime time , intel says , despite all the linux hype", "label":1}

{"source":"bowled by the slower one again , kolkata , november 14 the past caught up with sourav ganguly", "label":2}
```

Notes: Inference / Prediction

```
{
 "instances": ["the movie was excellent", "i did not like the plot ."]
}
```

Example:

- **Generating word vectors with BlazingText**

  https://github.com/aws/amazon-sagemaker-examples/blob/master/introduction_to_amazon_algorithms/blazingtext_word2vec_text8/blazingtext_word2vec_text8.ipynb

References:

1. https://aws.amazon.com/blogs/machine-learning/amazon-sagemaker-blazingtext-parallelizing-word2vec-on-multiple-cpus-or-gpus/
2. https://docs.aws.amazon.com/sagemaker/latest/dg/blazingtext.html

II . DeepAR Forecasting

Summary:

DeepAR is a supervised learning algorithm for “forecasting scalar (one-dimensional) time series using recurrent neural networks (RNN).” It outperforms classical forecasting methods such as ARIMA and ETS when the dataset contains hundreds of related time series. The algorithm accepts one or more target time series and trains a model to predict how the time series evolves.

Practical Problems it can solve:

1. Sales Forecasting
2. Webpage Request Forecasting

Terms and Definitions:

- **Autoregressive integrated moving average (ARIMA)**- a linear regression model that explains a given time series based on its own lags and lagged forecast errors.
- **Exponential smoothing (ETS)** - a time series forecasting method for univariate data.
- **Lags** - a delay or fixed amount of passing time.

Notes: Training

```
{"start": "2009-11-01 00:00:00", "target": [4.3, "NaN", 5.1, ...], "cat": [0, 1], "dynamic_feat": [[1.1, 1.2, 0.5, ..]]}
{"start": "2012-01-30 00:00:00", "target": [1.0, -5.0, ...], "cat": [2, 3], "dynamic_feat": [[1.1, 2.05, ...]]}
{"start": "1999-01-30 00:00:00", "target": [2.0, 1.0], "cat": [1, 4], "dynamic_feat": [[1.3, 0.4]]}
```

Notes: Inference / Prediction

```
{
   "num_samples": 100,
   "output_types": ["mean", "quantiles"],
   "quantiles": ["0.1", "0.2", "0.3", "0.4", "0.5", "0.6", "0.7", "0.8", "0.9"]
}
```

Example:

- **Time series forecasting with DeepAR - Synthetic data**

  https://sagemaker-examples.readthedocs.io/en/latest/introduction_to_amazon_algorithms/deepar_synthetic/deepar_synthetic.html

- **SageMaker/DeepAR demo on electricity dataset**

  https://sagemaker-examples.readthedocs.io/en/latest/introduction_to_amazon_algorithms/deepar_electricity/DeepAR-Electricity.html

References:

1. https://docs.aws.amazon.com/sagemaker/latest/dg/deepar.html
2. https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/
3. https://machinelearningmastery.com/exponential-smoothing-for-time-series-forecasting-in-python/

III . Image Classification

Summary:

Image Classification algorithm is a supervised learning algorithm that supports multi-label classification. It accepts an image as input and outputs one or more labels for that image. It utilizes a convolutional neural network (ResNet).

Practical Problems it can solve:

1. Detection of adult content in an image
2. Image classification

Terms and Definitions:

- **Convolutional neural network** - a class of deep neural network most commonly applied in visual analyzation.
- **ResNet** - a “residual neural network” that replicates how pyramidal cells in the cerebral cortex work by utilizing shortcuts to jump over layers.
- **RecordIO** - a set of binary data exchange formats.

Notes: Training

- **Train with Image Format**

```
5       1    your_image_directory/train_img_dog1.jpg
1000    0    your_image_directory/train_img_cat1.jpg
22      1    your_image_directory/train_img_dog2.jpg
```

- **Train with Augmented Manifest Image Format**

```
{"source-ref":"s3://image/filename1.jpg", "class":"0"}
{"source-ref":"s3://image/filename2.jpg",          "class":"1",
"class-metadata":     {"class-name":     "cat",     "type"     :
"groundtruth/image-classification"}}
```

Notes: Inference / Prediction

```
accept: application/jsonlines

 {"prediction": [prob_0, prob_1, prob_2, prob_3, ...]}
```

Example:

- **End-to-End Multiclass Image Classification**

  https://sagemaker-examples.readthedocs.io/en/latest/introduction_to_amazon_algorithms/imageclassification_caltech/Image-classification-fulltraining.html

References:

1. https://docs.aws.amazon.com/sagemaker/latest/dg/image-classification.html
2. https://docs.aws.amazon.com/sagemaker/latest/dg/algos.html
3. http://mesos.apache.org/documentation/latest/recordio/

IV . K-Means

Summary:

Image Classification algorithm is an unsupervised learning algorithm that attempts to find discrete groupings within data. It uses a modified version of the web-scale k-means clustering algorithm that is more accurate. Users define the attributes that the algorithm uses to determine similarity. Within data

Practical Problems it can solve:

1. Identifying crime localities
2. Customer segmentation

Terms and Definitions:

- **Euclidean distance** - the length of a line segment between two points.
- **Mini-batch k-means** - a modification to the k-means algorithm to address requirements for latency, scalability, and sparsity encountered in user-facing web applications.
- **Mini-batches** - small, random subsets of the training data.

Notes: Training

```
S3DataDistributionType=ShardedByS3Key)
recordIO-wrapped-protobuf
CSV
```

Notes: Inference / Prediction

- **JSONLINES Response Format**

```
{"closest_cluster": 1.0, "distance_to_cluster": 3.0}
{"closest_cluster": 2.0, "distance_to_cluster": 5.0}
```

- **RECORDIO Response Format**

```
[
```

```
    Record = {
        features = {},
        label = {
            'closest_cluster': {
                keys: [],
                values: [1.0, 2.0]  # float32
            },
            'distance_to_cluster': {
                keys: [],
                values: [3.0, 5.0]  # float32
            },
        }
    }
]
```

- **CSV Response Format**

```
1.0,3.0
2.0,5.0
```

Example:

- **Analyze US census data for population segmentation using Amazon SageMaker**

  https://sagemaker-examples.readthedocs.io/en/latest/introduction_to_applying_machine_learning/US-census_population_segmentation_PCA_Kmeans/sagemaker-countycensusclustering.html

References:

1. https://docs.aws.amazon.com/sagemaker/latest/dg/k-means.html
2. https://www.eecs.tufts.edu/~dsculley/papers/fastkmeans.pdf

V . Latent Dirichlet Allocation

Summary:

Latent Dirichlet Allocation algorithm is a built-in SageMaker unsupervised learning algorithm. It attempts to describe a set of observations as a mixture of distinct categories. The categories are not pre-set by the user and are not guaranteed to be aligned with human categorization.

Practical Problems it can solve:

1. Document classification
2. Customer clustering

Terms and Definitions:

- **Text corpus** - a language resource consisting of a large and structured set of texts.
- **Collapsed Gibbs sampling** - a sampler that integrates out or marginalizes over one or more variables when sampling for some other variable.
- **Generative probability model** - a model for the distribution of outputs and inputs based on latent variables.

Notes: Training

- **recordIO-wrapped-protobuf**
- **csv**

Notes: Inference / Prediction

- **text/csv**
- **application/json**
- **application/x-recordio-protobuf**

Example:

- **An Introduction to SageMaker LDA**

  https://sagemaker-examples.readthedocs.io/en/latest/introduction_to_amazon_algorithms/lda_topic_modeling/LDA-Introduction.html

References:

1. https://docs.aws.amazon.com/sagemaker/latest/dg/lda.html
2. https://docs.aws.amazon.com/sagemaker/latest/dg/lda-how-it-works.html

VI . Linear Learner

Summary:

The Linear Learner algorithm utilizes supervised learning linear models. These models are mainly used for either classification or regression problems. The main advantage of Linear Learner compared to similar algorithms is that it provides convenience and a significant increase in speed over hyperparameter optimization techniques.

Practical Problems it can solve:

1. Email spam filter
2. Real estate pricing

Terms and Definitions:

- **F1 measure** - measure of a test's accuracy.
- **Precision** - the ratio between true positives and all positives.
- **Recall** - measure of correctly identifying true positives.

Notes: Training

- **recordIO-wrapped-protobuf**
- **csv**

Notes: Inference / Prediction

- **JSONLINES Response Format (Binary Classification)**

```
{"score": 0.4, "predicted_label": 0}
```

- **JSONLINES Response Format (Multiclass Classification)**

```
{"score": [0.1, 0.2, 0.4, 0.3], "predicted_label": 2}
```

- **Regression**

```
{"score": 0.4}
```

Example:

- **Breast Cancer Prediction**

  https://sagemaker-examples.readthedocs.io/en/latest/introduction_to_applying_machine_learning/breast_cancer_prediction/Breast%20Cancer%20Prediction.html

- **Multiclass classifiers**

  https://sagemaker-examples.readthedocs.io/en/latest/scientific_details_of_algorithms/linear_learner_multiclass_classification/linear_learner_multiclass_classification.html

References:

1. https://docs.aws.amazon.com/sagemaker/latest/dg/linear-learner.html
2. https://www.analyticsvidhya.com/blog/2020/09/precision-recall-machine-learning/

VII . Object2Vec

Summary:

Object2Vec is a general-purpose neural embedding algorithm. It generalizes the Word2Vec embedding technique used in BlazingText. The embeddings can be used to compute nearest neighbors of objects or for supervised tasks such as classification or regression.

Practical Problems it can solve:

1. Identification of duplicate support tickets
2. Routing of support tickets based on similarity of text.

Terms and Definitions:

- **Neural embedding** - learned low-dimensional representations of discrete data as continuous vectors.
- **Dense embeddings** - vectors that are short and dense (most elements are non-zero).
- **Embeddings** - a relatively low-dimensional space into which you can translate high-dimensional vectors.

Notes: Training

- **Classification or Regression**

```
{"in0": [6, 17, 606, 19, 53, 67, 52, 12, 5, 10, 15, 10178, 7, 33, 652, 80, 15, 69, 821, 4], "in1": [16, 21, 13, 45, 14, 9, 80, 59, 164, 4]}
{"in0": [22, 1016, 32, 13, 25, 11, 5, 64, 573, 45, 5, 80, 15, 67, 21, 7, 9, 107, 4], "in1": [22, 32, 13, 25, 1016, 573, 3252, 4]}
{"in0": [774, 14, 21, 206], "in1": [21, 366, 125]}
```

- **Embeddings**

```
{"in0": [6, 17, 606, 19, 53, 67, 52, 12, 5, 10, 15, 10178, 7, 33, 652, 80, 15, 69, 821, 4]}
{"in0": [22, 1016, 32, 13, 25, 11, 5, 64, 573, 45, 5, 80, 15, 67, 21, 7, 9, 107, 4]}
```

```
{"in0": [774, 14, 21, 206]}
```


Notes: Inference / Prediction

- **Classification or Regression**


```
{
    "predictions": [
        {
            "scores": [
                0.6533935070037842,
                0.07582679390907288,
                0.2707797586917877
            ]
        },
        {
            "scores": [
                0.026291321963071823,
                0.6577019095420837,
                0.31600672006607056
            ]
        }
    ]
}
```


- **Embeddings**


```
{
  "predictions": [

{"embeddings":[0.057368703186511,0.030703511089086,0.099890425
801277,0.063688032329082,0.026327300816774,0.003637571120634,0
.021305780857801,0.004316598642617,0.0,0.003397724591195,0.0,0
.000378780066967,0.0,0.0,0.0,0.007419463712722]},

{"embeddings":[0.150190666317939,0.05145975202322,0.0982042700
05226,0.064249359071254,0.056249320507049,0.01513972133398,0.0
47553978860378,0.0,0.0,0.011533712036907,0.011472506448626,0.0
10696629062294,0.0,0.0,0.0,0.008508535102009]}
  ]
}
```

Example:

- **Encoding Sentences into Fixed Length Embeddings**

  https://sagemaker-examples.readthedocs.io/en/latest/introduction_to_amazon_algorithms/object2vec_sentence_similarity/object2vec_sentence_similarity.html

- **Learning document embeddings**

  https://sagemaker-examples.readthedocs.io/en/latest/introduction_to_applying_machine_learning/object2vec_document_embedding/object2vec_document_embedding.html

References:

1. https://docs.aws.amazon.com/sagemaker/latest/dg/object2vec.html
2. http://www.cs.umd.edu/class/fall2018/cmsc470/slides/slides_12.pdf
3. https://developers.google.com/machine-learning/crash-course/embeddings/video-lecture

VIII . Principal Component Analysis

Summary:

Principal Component Analysis is an unsupervised learning algorithm that attempts to reduce the dimensionality within a dataset. The new set of features are called components, which are composites of the original features. It has two modes: regular for datasets with sparse data and moderate number of features, and randomized for data with a large number of features.

Practical Problems it can solve:

1. Feature reduction
2. Conversion of high-dimensional into a 2D space.

Terms and Definitions:

- **Dimensionality** - the number of features a dataset has.
- **Covariance matrix** - square matrix that gives the covariance between each pair of elements of a given random vector.
- **Singular value decomposition** - an expansion of the original data in a coordinate system where the covariance matrix is diagonal.

Notes: Training

- **recordIO-wrapped-protobuf**
- **csv**

Notes: Inference / Prediction

```
{
    "projections": [
        {
            "projection": [1.0, 2.0, 3.0, 4.0, 5.0]
        },
        {
            "projection": [6.0, 7.0, 8.0, 9.0, 0.0]
        },
```

```
            ....
        ]
}
```

Example:

- **PCA with MNISt**

  https://sagemaker-examples.readthedocs.io/en/latest/introduction_to_amazon_algorithms/pca_mnist/pca_mnist.html

References:

1. https://docs.aws.amazon.com/sagemaker/latest/dg/pca.html
2. http://web.mit.edu/course/other/be.400/OldFiles/www/SVD/Singular_Value_Decomposition.htm

IX . Sequence-to-Sequence

Summary:

Sequence to Sequence is a built-in supervised learning algorithm. It accepts a sequence of tokens (e.g. text, audio) as input and generates another sequence of tokens as output. It uses RecRecurrenturennt Neural Networks and Convolutional Neural Networks.

Practical Problems it can solve:

1. Speech-to-text
2. Text summarization

Terms and Definitions:

- **Tokens** - an instance of a sequence of characters in some particular document that are grouped together as a useful semantic unit for processing.
- **Machine translation** - automated translation of text.
- **Speech-to-text** - conversion of real-time audio or audio files to text.

Notes: Training

```
{"source": "source_sequence_0"}
{"source": "source_sequence_1"}
```

Notes: Inference / Prediction

```
{"target": "predicted_sequence_0"}
{"target": "predicted_sequence_1"}
```

Example:

- **English-German Machine Translation**

  https://sagemaker-examples.readthedocs.io/en/latest/introduction_to_amazon_algorithms/seq2seq_translation_en-de/SageMaker-Seq2Seq-Translation-English-German.html

References:

1. https://docs.aws.amazon.com/sagemaker/latest/dg/seq-2-seq.html
2. https://nlp.stanford.edu/IR-book/html/htmledition/tokenization-1.html
3. https://lingohub.com/academy/glossary/machine-translation

X . XGBoost

Summary:

The built-in SageMaker XGBoost or eXtreme Gradient Boosting algorithm is based on the open-source implementation of gradient boosted trees algorithm. It is a supervised learning algorithm that predicts that target variable by combining an ensemble of estimates from simpler models. It is used for regression, classification, and ranking problems.

Practical Problems it can solve:

1. House value estimation
2. Customer churn prediction

Terms and Definitions:

- **Gradient boosted trees** - a machine learning technique for optimizing the predictive value of a model through successive steps in the learning process.
- **L1 Regularization** - regularization method that adds “absolute value of magnitude” of coefficient as penalty term to the loss function.
- **L2 Regularization** - regularization method that adds “squared magnitude” of coefficient as penalty term to the loss function.

Notes: Training

```
label:weight idx_0:val_0 idx_1:val_1..
```

Notes: Inference / Prediction

- **libsvm**
- **csv**

Example:

- **Customer Churn Prediction**

  https://sagemaker-examples.readthedocs.io/en/latest/introduction_to_applying_machine_learning/xgboost_customer_churn/xgboost_customer_churn.html

- **Regression with XGBoost**

  https://sagemaker-examples.readthedocs.io/en/latest/introduction_to_amazon_algorithms/xgboost_abalone/xgboost_parquet_input_training.html

References:

1. https://docs.aws.amazon.com/sagemaker/latest/dg/xgboost.html
2. https://c3.ai/glossary/data-science/gradient-boosted-decision-trees-gbdt/
3. https://towardsdatascience.com/l1-and-l2-regularization-methods-ce25e7fc831c