Algorithm	Algo. Type	Input Format	INT/FLOAT	Processor Type	Instance	Multiprocessor in Single Machine	Multi Machine	Use Cases	Comments	HP
Linear Learner	SUPERVISED	* RecordIO **Wrapped** Protobuf / CSV * Float32 Data only	FLOAT32	CPU GPU	Any	CPU GPU	Only CPU No GPU	* Regression and classification * Classification: Binary or multiclass	* Need data to be normalized else algo may not converge * Multiple models are trained in parallel	**balance_multiclass_weights** learning_rate mini_batch_Size L1, L2
XGBoost	SUPERVISED	* CSV or LibSVM (not AWS algo, but adapted, hence NO RecordIO-protobuf)	-	CPU	M4	-	No	* Regression and classification * Classification: Binary or multiclass	* Output model as pickle * Uses extreme boosting of trees * Algo is memory bound, not much compute	* **subsample_trees** (less overfitting) * eta (eq. to learning rate) * alpha, gamma, lambda (conservative trees for higher values)
Seq2Seq	SUPERVISED	* RecordIO-Protobuf	INT	GPU	P3	GPU	No	* Machine translation * Text summarization * Speech to text * Any use case where input a sequence and output is a sequence	* Along with training data and validation data files, must provide **vocabulary files -- in case of text seq2seq** * Start with tokenized text files, then convert to RecordIO-Protobuf * Uses RNNs and CNNs internally	* batch_size * optimizer * learning_rate *** num_layers_encoder** *** num_layers_decoder** * can optimize on: accuracy, BLEU score (mach. translation), perplexity
DeepAR	SUPERVISED	* JSON Lines * GZIP * Parquet -- Each record to contain - **Start**: starting TS - **Target**: the TS values to learn/predict	-	CPU GPU	C4 P3	CPU GPU	CPU GPU	* Stock price prediction * Sales and promotion effectiveness * Any time oriented forecasting, single dimension	* Uses **RNNs** * Can train several related timeseries, more series the better results, learns relationships b/w timeseries * Start with CPU (C4.2xlarge, or higher), if necessary, move to GPU. Only large models need GPU	* **context_length** (number of time points back in time the model learns) * epochs, batch_size, learning_rate, num_cells
Blazing Text - Text Classification	SUPERVISED	Augmented manifest text format -- "__label__1 this is a sentence with , punctuations also tokenized . that is space delimited . One sentence per line . label at the start"	-	CPU GPU	size < 2GB: C5 size > 2GB: P2, P3	Single GPU	No	* web search and information retrieval	* predict labels for sentence	* epochs * learning_rate * word_ngrams * **vector_dim**

Algorithm	Algo. Type	Input Format	INT/FLOAT	Processor Type	Instance	Multiprocessor in Single Machine	Multi Machine	Use Cases	Comments	HP
Blazing Text - Word2Vec	UNSUPERVISED	**Word2Vec** one sentence per line	-	CPU GPU	P3	CPU/GPU: CBOW & Skip Gram	GPU: Batch skip gram CPU: No	* Preparing input for NLP use cases * Vectorization of text for machine translation and sentiment analysis * Semantic similarity of words	* Represents words as vectors * Semantically similar words are represented by vectors close to each other * Semantic -- of or relating to meaning in language **MULTIPLE MODES:** * ***CBOW*** - Continuous Bag of Words - Order of words DO NOT matter * ***Skip Gram*** i.e. n-gram - order of words matter * ***Batch skip gram*** - order of words matter	* mode: mandatory * learning_rate * window_size * **vector_dim** * negative_Samples
Object2Vec		* Any object to be tokenized into integers * Training data: - pairs of tokens - sequence of tokens	INT	CPU GPU	M5, P2	Single machine	No	* Collaborative recommendation system * Multi-label document classification system * Sentence Embeddings * Learns relations or associations: - sen to sen - labels to seq (genre to description) - product to product (recommendation) - user to item (recommendation)	* CNNs and RNNs used * Encoders used in input - uses 2 encoders in parallel - learns associations b/w encoders, using a comparator **Encoder types:** * Hierchical CNNs (hCNNs) * bi-lstm * pooled_embedding	dropout, early_stopping_epochs, learning_rate, batch_size, layers, act. func., optimizer, weight_decay
Object Detection	SUPERVISED	RecordIO (NOT Protobuf) or Images (JPEG or PNG) + With image manifest in JSON, one JSON per image that contains annotations	-	GPU	P2, P3	Yes	Yes	* Detect objects in an image * Object tracking	* Uses CNN with SSD * Transfer learning/incremental learning supported * Uses FLIP, RESCALE, JITTER internally to avoid overfitting * CPUs can be used for inference, not for training	Standard CNN HPs like: learning_rate, batch size, optimizer etc.
Image Classification	SUPERVISED	* Pipe: Apache MxNET RecordIO (NOT Protobuf) - for interoperability with other DNN frameworks * File Mode: Raw JPEG/PNG + *.LST files - associates image index, class label, path to image -- To use images directly in Pipe mode use JSON based Augmented Manifest Format	-	GPU	P2, P3	Yes	Yes	* classify images into multiple classes * dog/cat/rat/tiger etc.	* **Full training**: ResNet CNN is used. N/W initialized with random weights * **Transfer Learning/Pre-trained**: Image Net is used. Initialized with pre trained weights. Only Top FC layer is initialized with random weights. * CPU can be used for inference, if not suitable, move to GPU	* batch_size * learning_rate * optimizer, B1, B2, eps, Gamma

Algorithm	Algo. Type	Input Format	INT/FLOAT	Processor Type	Instance	Multiprocessor in Single Machine	Multi Machine	Use Cases	Comments	HP
Semantic Segmentation	SUPERVISED	* Raw JPEG/PNG in file mode + annotations * Add Augmented Manifest Format for Pipe Mode	-	GPU	P2, P3	Yes	No	* Self driving cars * Medical imaging and diagnostics * Robot sensing * Given a pixel - what object does it belong to ?	* Algo under hood: Gluon CV of MxNET = FC + Pyramid Scene Pairing + DeepLabV3 * Arch: **ResNet50/ResNet101** = "**Backbone**" selection in HP * Trained on ImageNet data * Incremental/Transfer learning allowed * Inference can use CPU or GPU Each of the three algorithms has two distinct components: * The **backbone (or encoder)**—A network that produces reliable **activation maps** of features. * The **decoder**—A network that constructs the **segmentation mask from the encoded activation maps**. The segmentation output is represented as a **grayscale** image, called a **segmentation mask.** A segmentation mask is a grayscale image with the **same shape as the input image**.	epochs, learning_rate, batch size, algo, **backbone**
Random Cut Forest	UNSUPERVISED	* RecordIO-Protobuf * CSV	-	CPU	M4,C4,C5	-	No	* Anomaly detection * Detect unexpected spikes in TS data * Few people have tried using this for fraud detection	* Assigns anomaly score to each data point * Uses forest of trees * Looks at **expected change in complexity as a result of adding a point to a tree** * Random sampling * RCF is used in Kinesis Analytics in real time	num_trees, **num_samples_per_tree** (= choose inversely proportional to ratio #anomalous/#normal in dataset)
Neural Topic Modelling	UNSUPERVISED	* RecordIO-Protobuf * CSV - Words must be tokenized to integers - aux channel for vocab	INT	GPU	P2, P3	-		* Organize docs into topics * Summarize docs based on topics	* Algo: **Neural Variational Inference** * Define how many topics to group docs into * Used only on text * CPU / GPU for inference	**num_topics** mini_batch_size learning_rate **variation_loss** (at expence of learing time)
LDA (Latent Dirichlet Allocation)	UNSUPERVISED	* RecordIO-Protobuf (Pipe Mode) * CSV - Words must be tokenized to integers - aux channel for vocab	-	CPU	M4	No	No	* Cluster customers based on purchases * Harmonic analysis in music	* Algo: LDA - Open source availability, **not DNN** * Can process more than text, like harmonic music analysis * Single inst. CPU	**num_topics** **alpha0** = small values - sparse topic mixtures, >1 uniform topic mixture

Algorithm	Algo. Type	Input Format	INT/FLOAT	Processor Type	Instance	Multiprocessor in Single Machine	Multi Machine	Use Cases	Comments	HP
kNN (k Nearest Neighbors)	SUPERVISED	* RecordIO-protobuf * CSV -- File or pipe mode both - first column has label	-	CPU GPU	-	-	-	* Classification and regression	* Sagemaker automates 3 steps: - Sample data (can't use for huge data) - Dim reduction (sign or nfjlt methods) - Build index for looking up neighbours	**k** **sample_size**
K-Means	UNSUPERVISED	* RecordIO-protobuf * CSV -- File or pipe mode both	-	CPU (recommended) GPU	M4, M5, C4, C5	-	-	* Cluster data - unsupervised * Find groups of data points based on similarity	* Webscale K-Means in Sagemaker * Similarity measured by euclidean distance * Works to optimize the centers of eack of the k-clusters * **Algorithm**: 1) Determine init. cluster centers = 2 ways: **k-means++** (tries to make initial clusters far apart) OR **random** 2) Iterate over data and calculate cluster center 3) Reduce from K to k - using Lloyd's method or k-means++ K comes from "extra_cluster_centers" which improves accuracy, but later reduced to k. K = k * x	* **K** * mini_batch_size * **extra_center_factor (x)** * **init_method** (k-means++ OR random)
PCA - Principal Component Analysis	UNSUPERVISED	* RecordIO-protobuf * CSV -- File or pipe mode both	-	CPU GPU	-	-	-	* Dimensionality Reduction * Removes Curse of Dimensionality	* Reduced Dimensions are called components * 1st component - largest possible variaility, next 2nd component, so on .. * Used Singular Value Decomposition (SVD) * **Two Modes**: - **Regular**: Sparse data. modelate #features, #rows - **Randomized**: Dense data. #large data, #large features, uses approximation algos	* **algorithm_mode** (regular, random * **subtract_mean**: unbiases data
Factorization Machines	SUPERVISED	* RecordIO-Protobuf	FLOAT32	CPU (recommended) GPU	-	-	-	* Regression, Classification, recommendation - all in one general purpose algo for sparse data * Click prediction * Item recommendation	* Limited to pairwise interaction - 2nd order e.g. user to item interactions * **CSV not practical** hence not supported,a s data is sparse ***GPU not recommented** as data is sparse, GPU works better on dense data	* **Initialization methods** for bias, factors and linear terms - methods: uniform, normal or const - can tune properties of each method

Algorithm	Algo. Type	Input Format	INT/FLOAT	Processor Type	Instance	Multiprocessor in Single Machine	Multi Machine	Use Cases	Comments	HP
IP Insights	UNSUPERVISED	* CSV only for training * Inference: JSON lines, CSV, JSON	-	CPU GPU (recommended)	-	Multi GPU	-	* Identify suspicious IP addresses in context of security * Logins from anomalous IPs * Identify accounts creating resources from anamolous IPs	* **Only IPv4 supported** * Uses **NN** to learn latent vector rep. of entities and IP addresses * entities are hashed and embedded - large hash size * Automatically generates anomalous data by randomly pairing entities and IPs - as data will be highly imbalanced	* **num_entity_vectors** (hash size, set to twice the unique entity identifiers) * **vector_dim** (size of embedding vectors, scales model size) * Others: epoch, batch_size, leraning_rate, etc.
Reinforcement Learning	REINFORCEMENT LEARNING	* Nothing specific to Sagemaker	-	GPU	GPU	Yes	Yes - Multi Instance GPU recommended	* Games * Supply chain management * HVAC Systems * Industrial robotics * Dialog systems * Autonomous vehicles	* Supports Intel coach, Ray RLLib * Tensorflow, MxNET * Custom, commercial and opensource environments supported - Matlab simulink, energy plus, robo school, pybullet, Amazon Sumerian, AWS Robomaker	* Depends on framework and algo used, nothing tied to Sagemaker

Mandatory FLOAT32	Mandatory INT32	CPU ONLY	GPU Only	Incremental Training Available
Linear Learner	Seq2Seq	XGBoost	Seq2Seq	Image Classification
Factorization Machines	Object2Vec	RCF	Image Classification	Semantic Segmentation
	NTM	LDA	Semantic Segmentation	Object Detection
			Object Detection	
			NTM	
			RL	