BigML Deepnet Cheat Sheet

Deepnet Configuration

Deepnet Configuration Options

Option	Description	Default	API Name
Objective field	The field you want to predict. It can be a categorical or numeric field.	Last valid field in dataset	objective_field
Maximum Training Time	The maximum wall-clock time in minutes to train the deepnet. Alternatively, you can set a maximum number of iterations to limit the deepnet runtime.	1,800	max_training_time
Maximum Iterations	The maximum number of gradient steps the algorithm takes during the optimization process. You can set from 100 up to 100,000 maximum iterations. Alternatively, you can set a maximum training time to limit the deepnet runtime.	20,000	max_iterations
Default numeric value	Replaces missing numeric values in your dataset by the field's maximum, mean, median, minimum, or zero. If you do not activate this option or **Missing numerics** option, your instances with missing numeric values will be ignored.	Null	default_numeric_value
Missing numerics	Allows the deepnet to consider missing values for the numeric fields as valid values. If you do not activate this option or set a **Default numeric value**, your instances with missing numeric values will be ignored.	True	missing_numerics

Automatic Optimization Options

Option	Description	Default	API Name
Automatic Network Search	During the deepnet creation, BigML trains and evaluates over all possible network configurations, returning the best networks found for your problem. The final deepnet returned by the search is a "compromise" between the top "n" networks found in the search. Since this option builds several networks, it may be significantly slower than the structure suggestion technique.	False	search
Automatic Structure Suggestion	A faster alternative to the search technique that can still produce high quality networks. BigML has learned some general rules about what makes one network structure better than another for a given dataset. BigML will automatically suggest a structure and a set of parameter values that are likely to perform well for your dataset. This option only builds one network.	True	suggest_structure

Network Architecture

Option	Description	Default	API Name
Hidden layers	Sets the intermediate layers between the input layer (containing the input field values) and the output layer (containing the predictions). Each layer is composed of several nodes that transform the raw input values using an activation function to represent a non-linear mapping between the inputs and the outputs. You can set the total number of layers up to 32, select the activation function (Tanh, Sigmoid, Softplus, Softmax, ReLU or no function) and the number of nodes per layer.	Null	hidden_layers
Learn residuals	Tweaks the mathematical formula of each layer's equation to include the inputs of a lower layer in a node of a higher layer. It will cause alternate layers to learn a representation of the residuals for a given layer rather than the layer itself, by introducing shortcut connections.	False	learn_residuals
Batch normalization	Allows the outputs of a network to be normalized before being passed to the activation function. This will introduce extra parameters in each layer (the mean, variance, and scale of the layer), and will significantly slow down training.	False	batch_normalization
Tree Embedding	Learns a tree-based representation of the data as engineered features along with the raw features, essentially by learning trees over slices of the input space. The theory is that these engineered features will linearize obvious non-linear dependencies before training begins accelerating the learning process.	False	tree_embedding

Algorithm Options

Option	Description	Default	API Name
Gradient Descent Algorithm	Sets the optimization algorithm used to minimize the loss function. You can select among several algorithms: Adam, Adagrad, Momentum, RMS Prop, FTRL. If your data is sparse, some of the adaptive algorithms (i.e., the ones that adapt the learning rate to each parameter may perform better (Adagrad, RMS Prop, and Adam).	Null	optimizer
Seed	Sets the random seed controlling the ordering of training data, the initial network weights, and the behavior of dropout during training. By setting the same seed you can get repeatable deepnets using the same dataset.	Null	deepnet_seed
Learning rate	Controls how aggressively the gradient descent algorithm fits the training data. You can set values greater than 0% and smaller than 100%. Larger values will prevent overfitting (usually 1% or lower), although it usually takes longer to train the deepnet.	1%	learning_rate
Dropout rate	Sets the proportion of nodes dropped from the network at training time. This prevents nodes from co-adapting so it is an effective method to control overfitting.	0%	dropout_rate
Beta1	Sets decay rate for the first moment estimate for the Adam algorithm (the mean).	0.9	beta1
Beta2	Sets decay rate for the second moment estimate for the Adam algorithm (the variance).	0.999	beta2
Epsilon	Sets a small constant for numerical stability. It is used in Adam and RMS Prop algorithms.	1.0e-08	eps
Momentum	Sets the acceleration of the gradient descent. Higher values accelerate the gradient descent. It is used in RMS Prop and momentum algorithms.	0.9	momentum
Initial accumulator value	Sets the initial value for the gradient accumulator. It is used in FTRL and Adagrad algorithms.	0.1	initial_accumulator_value
Decay	Sets the speed decay the moving average. It is used in RMS Prop algorithm.	0.1	decay
Regularization	Sets the regularization factor to avoid overfitting, i.e., tailoring the model to the training data at the expense of generalization. You can choose between L1 or L2 regularization. It is used in FTRL algorithm.	L1	regularization
Strength	Sets the regularization strength, so higher values indicate more regularization. It must be a positive integer greater than 0. It is used in FTRL algorithm.	1	c
Learning rate power	Sets the learning rate power of the FTRL algorithm. It needs to be a real number less or equal to zero.	-0.5	learning_rate_power

Output File Options

Option	Description	Default	API Name
Fields separator	Allows you to choose the best separator for your fields.	Comma	separator
New line	Sets the character to use as the line break in the generated csv file: "LF", "CRLF".	LF	newline
Show/hide fields	Allows you to show or hide the rest of the fields in your output file.	True	output_fields
Headers	Allows you to show or hide the names of your columns in the output file.	True	header
Prediction column name	Allows you to set the name you want for the objective field. By default BigML takes the name of the deepnet objective field.	Objective Field Name	prediction_name
Include probability	Allows you to include an additional column with the probability of the predicted class per instance.	False	probability
Probability column name	Allows you to set the name you want for the probability column.	Probability	probability_name
All class probability	Allows you to include all class probabilities per instance.	False	probabilities

Output Dataset

Option	Description	Default	API Name
Output dataset	Defines whether a dataset with the results should be automatically created or not.	True	output_dataset

⚙ Prediction Configuration

Probability threshold

Option	Description	Default	API Name
Probability threshold	A percentage between 0% and 100% that is commonly used with unbalanced classification models. When the probability of the positive class is above the established threshold, the non-positive class is predicted instead.	Null	operating_point

Default Numeric Values

Option	Description	Default	API Name
Default numeric value	Replaces missing numeric values in your dataset by the field's maximum, mean, median, minimum, or zero. If you do not activate this option and you did not trained your deepnet with Missing numerics, your instances with missing numeric values will be ignored and you will not get a prediction for them.	Null	default_numeric_value

Weights

Option	Description	Default	API Name
Balance objective	Sets instance weights so that each class in the objective field has equal influence on the deepnet.	False	balance_objective
Objective weights	Sets a specific weight for each class of the objective field. If a class is not listed, it is assumed to have a weight of 1. Weights of 0 are also valid.	False	objective_weights
Weight field	Sets instance weights using the values of the given field. The selected field must be numeric and it must not contain missing values. This is valid for both regression and classification deepnets.	False	weight_field

Sampling

Option	Description	Default	API Name
Rate	Sets the proportion of the dataset you want to consider between 0% and 100%.	100%	sample_rate
Range	Specifies a subset of instances from which to sample, e.g., from instance 5 to instance 1,000. The Rate you set will be computed over the Range configured.	(1, max. rows in dataset)	range
Sampling	Allows you to choose between a random sampling or a deterministic sampling. When using deterministic sampling the random-number generator will always use the same seed, producing repeatable results.	Random	seed
Replacement	Allows a single instance to be selected multiple times. Sampling without replacement ensures that each instance cannot be selected more than once.	False	replacement
Out of bag	Selects only the out-of-bag instances for the currently defined sample. If an instance is not selected as part of a sample, it is considered out of bag. It is only selectable when a sample is deterministic and the sample rate is less than 100%.	False	out_of_bag