Data Mining & Machine Learning F20DL

Group 4
Lewis Wilson, Sam Fay-Hunt, Kamil Szymczak, Chun Man

December 5, 2020

Contents

1 Introduction

For both decision trees and neural networks, we used Weights and Biases (`https://wandb.ai/home`) to visualise these experiments which we used to record and organise the experiment data shown the graphs shown in this document.

2 Variation in performance with size of the training and testing sets

3 Variation in performance with the change in the learning paradigm (Decision Trees versus Neural Nets)

4 Variation in performance with varying learning parameters in Decision Trees

4.1 J48

max_depth - The maximum depth of the tree.
min_impurity_decrease - Splits a node if this split induces a decrease of the impurity greater than or equal to this value.
min_samples_leaf - the minimum number of samples to be considered a leaf.
min_weight_fraction_leaf - The minimum number of samples required to be at a leaf node.
criterion_value -['Gini','Entropy']: Measures the quality of the split. **max_features -** ['auto','log2','sqrt'] : max features considers the number of features when looking for the best split. Auto just picks the best result so had the same result as log2.
Splitter - ['best', 'random']:

Parameter	**Type**	**Conclusion**
max_depth	int	had low importance value of 0.012 and provided some correlation 0.084.
min_impurity_decrease	float	had a low importance value of 0.051 and a strong negative correlation of -0.241.
min_samples_leaf	int or float	had low importance 0.014 and provided a tiny positive correlation of 0.007.
min_weight_fraction_leaf	float	gives a strong negative correlation in terms of accuracy, meaning the higher the min_weight_fraction_leaf value, the lower the accuracy.
criterion	string	Both provided a very low importance value of 0.003. Entropy had a very small negative correlation of -0.022 and Gini a positive correlation of 0.022.
max_features	string	Log2 provided an importance of 0.133 and a high negative correlation of -0.372. Log2 provided a tiny importance of 0.003 but a relatively good correlation of 0.196.
splitter	string	'best' and 'random' with best-having importance of 0.048 and random 0.044. Best has a positive correlation of 0.051 whereas random -0.051.

4.2 Random Forest

min_samples_split - The minimum number of samples required to split an internal node
min_samples_leaf - The minimum number of samples required to be at a leaf node.
n_estimators - The number of trees in the forest.
min_weight_fraction_leaf - The minimum number of samples required to be at a leaf node.
max_features - The number of features to consider when looking for the best
criterion - The function to measure the quality of a split. Supported criteria are "gini" for the Gini impurity and "entropy" for the information gain.

Parameter	**Conclusion**
max_features	Contains the options "auto", "sqrt" and "log2". From (Figure 3) we can see that "sqrt" has a higher accuracy overall, the accuracy of "log2" varies between the lower end and the median accuracy value.
min_samples_split	The minimum samples required to split a node has very little impact on accuracy.
criterion	Gives a perfect negative correlation with respect to accuracy. Correlation values being [Gini = -0.404], [Entropy = 0.404].
n_estimators	This is defined as the number of trees in the forest, it seems to have very little correlation but high importance.
min_samples_leaf	Gives a strong negative correlation in terms of accuracy, meaning the higher minimum samples at a leaf node, the lower the accuracy.
min_weight_fraction_leaf	Has a somewhat positive correlation to accuracy. e.g. total weight required at a leaf node varies between 76% and 89% accuracy

See Parameter Importance (Figure C.1)

5 Variation in performance with varying learning parameters in Neural Networks

5.1 Linear Classifier

For a linear classifier we have decided to use logistic regression. Shown below is the hyperparameter importance concerning the performance of Logistic Regression.

The accuracy fluctuates depending on which hyperparameters are used. Logistic Regression has a 'solver' hyperparameter in sklearn which is the algorithm to use in the optimization. We have tested the following: 'newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga'. Sag and Saga give the best results with 89Other solvers have a negative impact on the performance, this can be seen in the Solver parameter importance table where their correlation values are negative.

See Parameter Importance (Figure D.1)

Additionally 'sag' and 'saga' guarantee fast convergence on features with approximately the same scale [R41] which is the case with our datasets as pixel values have the same scale. This can be seen as the number of iterations between 200-800 don't change much in the accuracy and from the scatter chart of the accuracy of the Logistic Regression vs the number of sweeps created.

Parameter	**Conclusion**
C	The smaller this value is the stronger the regularization. For this we settled on 0.08027. Although this parameter did not make a big difference and provided similar results for values between 0.8 and 1.2
fit_intercept	From testing we have found out that for this dataset adding a bias works better than not having bias thus True
max_iter	We settled on 342 maximum number of iterations taken for the solver to converge. This parameter does not have a big importance but we found 342 gives accurate results.
solver	Contains the options 'newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga', default='lbfgs'. 'Saga' gives the best accuracy overall. Solver is the most influential hyperparameter as we can see from the importance and correlation table
tol	tolerance for stopping criteria which tells the algorithm to stop searching when some tolerance is achieved. This parameter did not make a difference, we settled on 0.0002277

5.2 Multilayer Perceptron

Parameter	**Conclusion**
alpha	This has a positive correlation to accuracy as higher alpha value equates to higher accuracy.
solver	lbfgs is the most accurate value of this parameter with a strong positive correlation out of the three (lbfgs, adam, sgd).
max_iter	The maximum number of iterations - In general, higher accuracy can be achieved with a larger amount of max iterations.
activation	Out of the four activation functions (relu, tanh, identity and logistic), relu is the only one with a positive correlation, giving the highest accuracy overall.
learning_rate	'adaptive' achieves the highest accuracy while, 'constant' and 'invscaling' vary widely.
hidden_layer_sizes	Has a negative correlation - the number neurons in the n-th hidden layer has no effect on accuracy.

6 Variation in performance according to different metrics (TP Rate, FP Rate, Precision, Recall, F Measure, ROC Area)

Appendices

A Appendix A

A.1 Workload split

Team member	Involvement
Lewis Wilson	text here
Chun Man	text here
Sam Fay-Hunt	text here
Kamil Szymczak	text here

As a team we are happy with everyone's contributions to the project. All team members were punctual and showed up to all scheduled meetings. Sam took the lead as project manager throughout the project delegating the workload and providing support to others.

B J48

J48 PARAM TABLE HERE

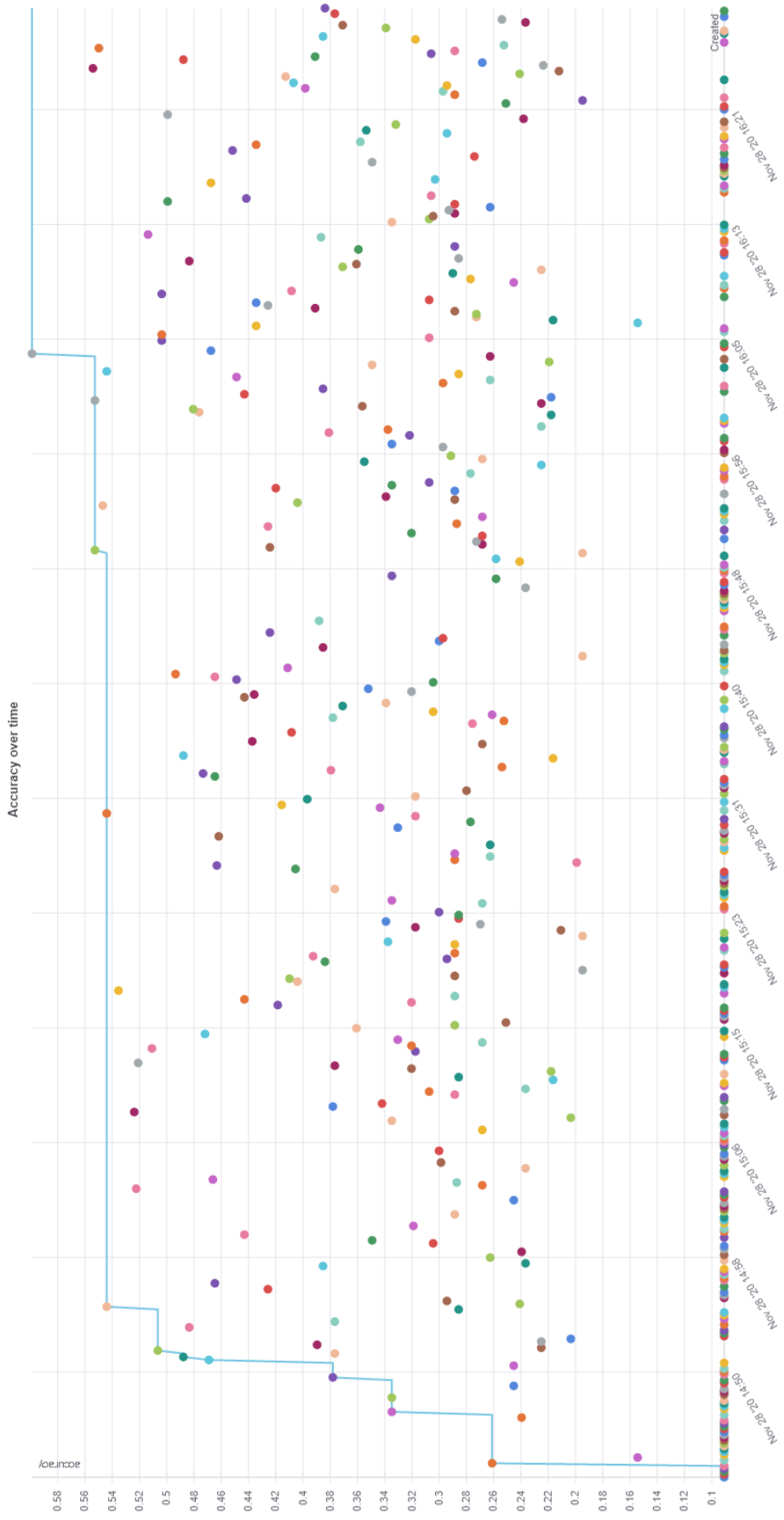

Figure 1: Accuracy over time - J48

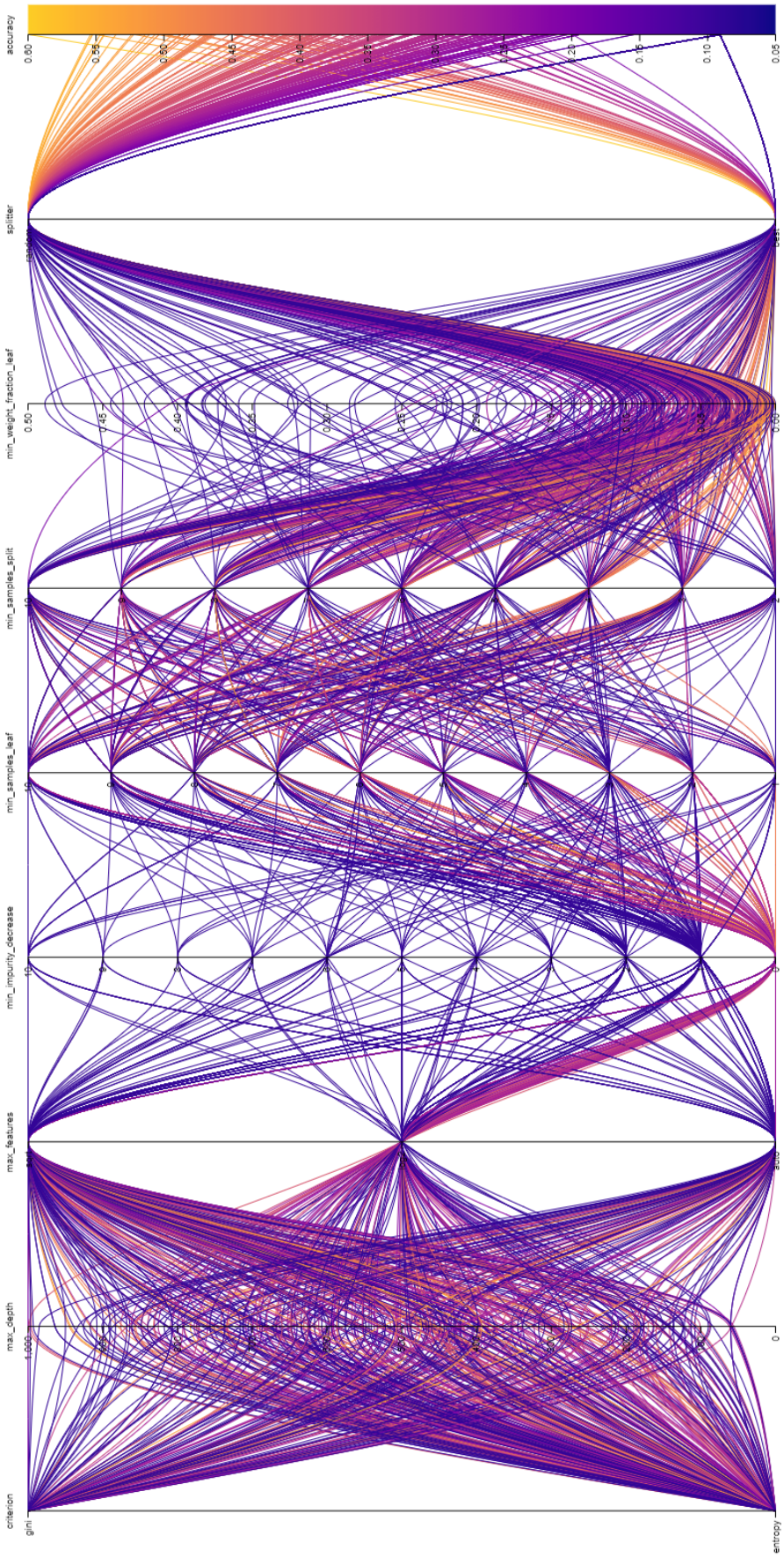

Figure 2: J48 Parameters

C Random Forest

C.1 Random Forest Parameter Importance

Parameter Config	Importance	Correlation
min_samples_split	0.005	0.306
min_samples_leaf	0.375	-0.725
n_estimators	0.016	0.092
min_weight_fraction_leaf	0.013	0.123

Parameter Config	Importance	Correlation
max_features.value_sqrt	0.001	0.563
max_features.value_log2	0.565	-0.752
criterion.value_entropy	0.012	0.404
criterion.value_gini	0.012	-0.404

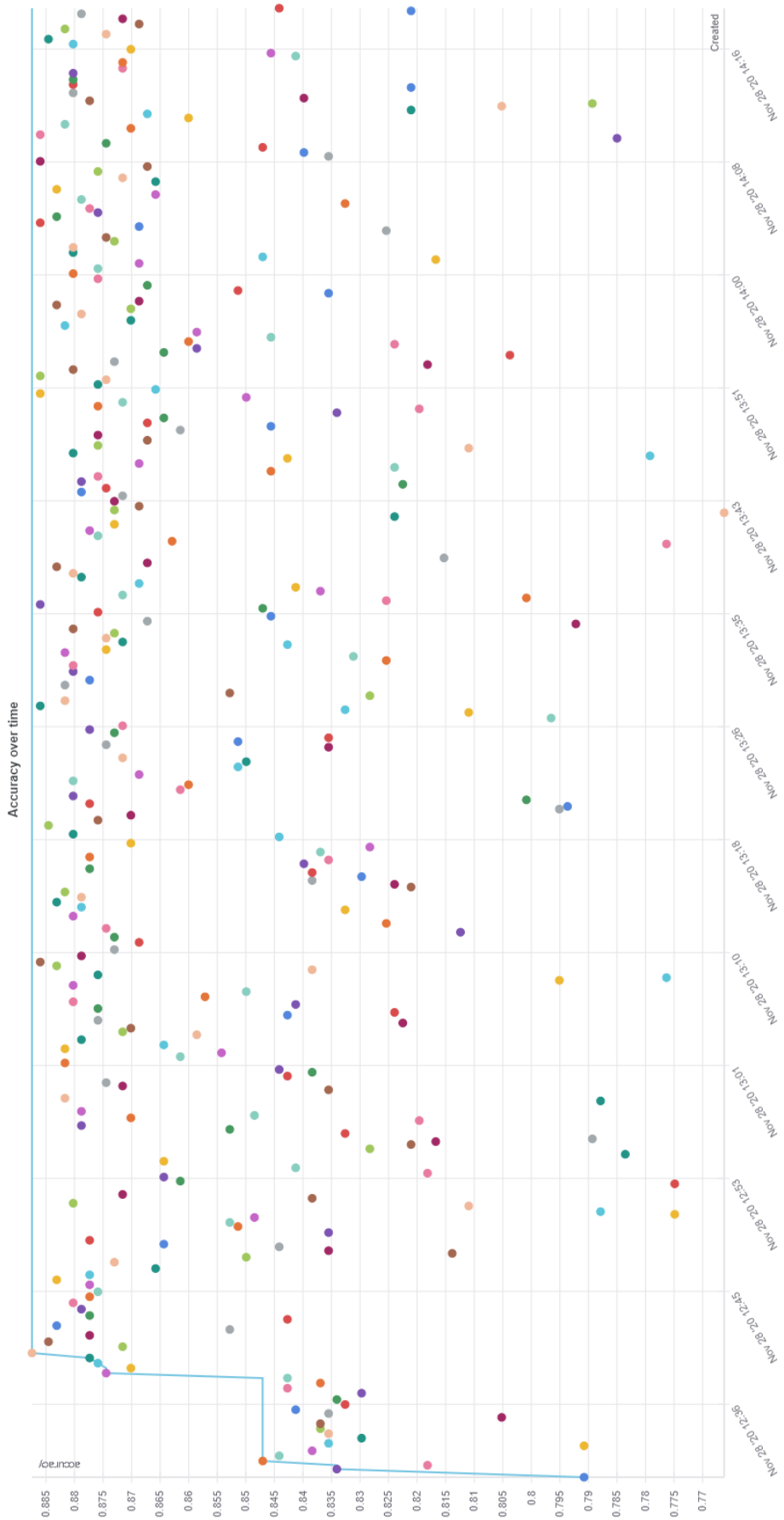

Figure 3: Accuracy over time - Random Forest

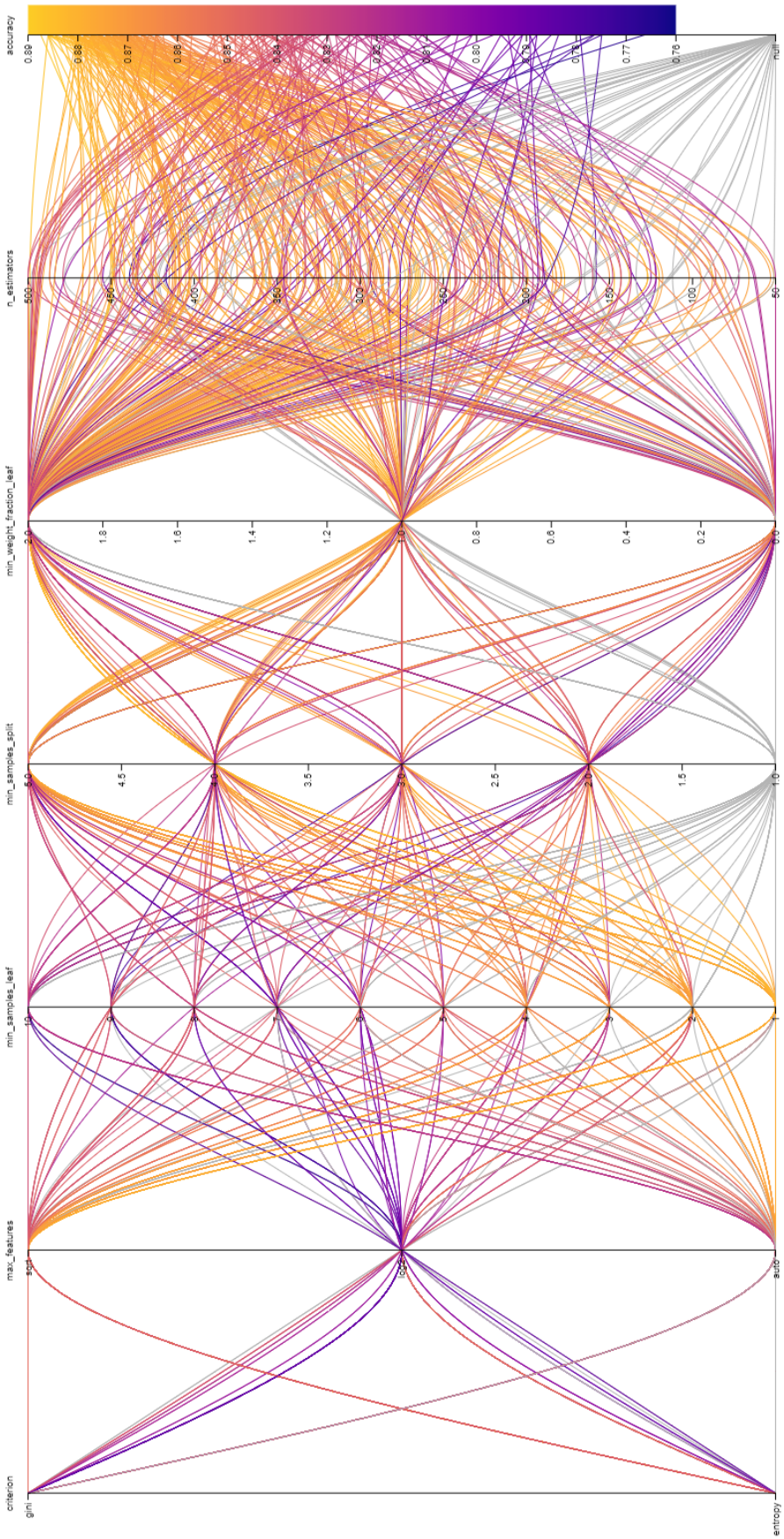

Figure 4: Random Forest Parameters

D Linear Classifier

D.1 Linear Classifier Parameter Importance

Solver Parameter Config	Importance	Correlation
lbfgs	0.786	-0.869
newton-cg	0.159	-0.271
liblinear	0.042	-0.037
saga	0.009	0.691
sag	0.003	0.170

Config Parameter Config	Importance	Correlation
max-iter	0.194	-0.634
l1-ratio	0.124	-0.510
fit-intercept	0.058	0.367
tol	0.046	-0.174
C	0.031	0.260

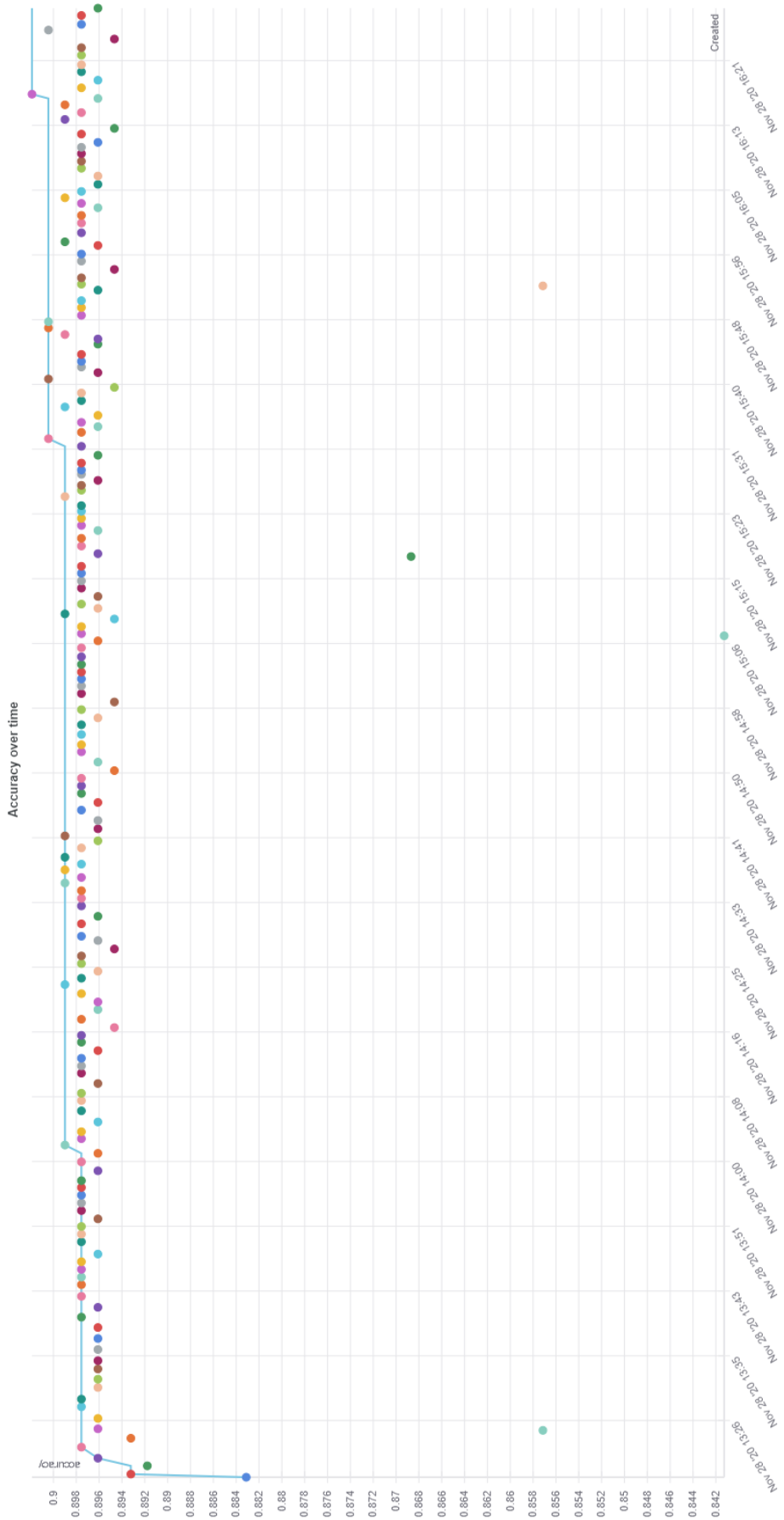

Figure 5: Accuracy over time - Linear Classifier

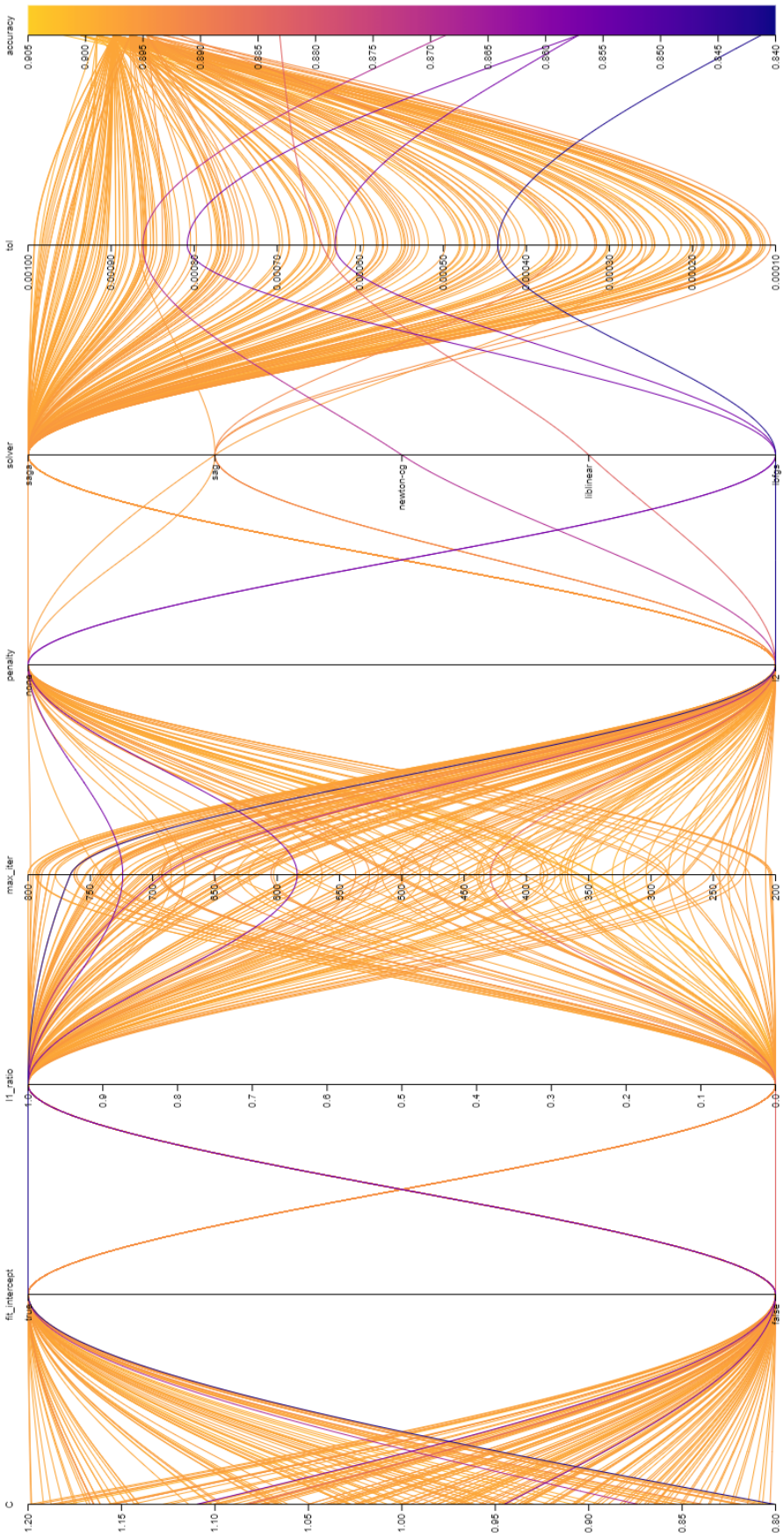

Figure 6: Random Forest Parameters

E Multilayer Perceptron

E.1 Multilayer Perceptron Parameter Importance

Parameter Config	Importance	Correlation
hidden_layer_sizes	0.095	-0.101
max_iter	0.091	0.072
alpha	0.061	0.171

Parameter Config	Importance	Correlation
solver.value_lbfgs	0.530	0.728
solver.value_adam	0.027	-0.245
solver.value_sgd	0.024	-0.640
activation.value_identity	0.098	-0.169
activation.value_relu	0.033	0.301
activation.value_tanh	0.018	-0.035
activation.value_logistic	0.006	-0.270
learning_rate.value_adaptive	0.012	0.507
learning_rate.value_constant	0.004	-0.442
learning_rate.value_invscaling	0.001	-0.224

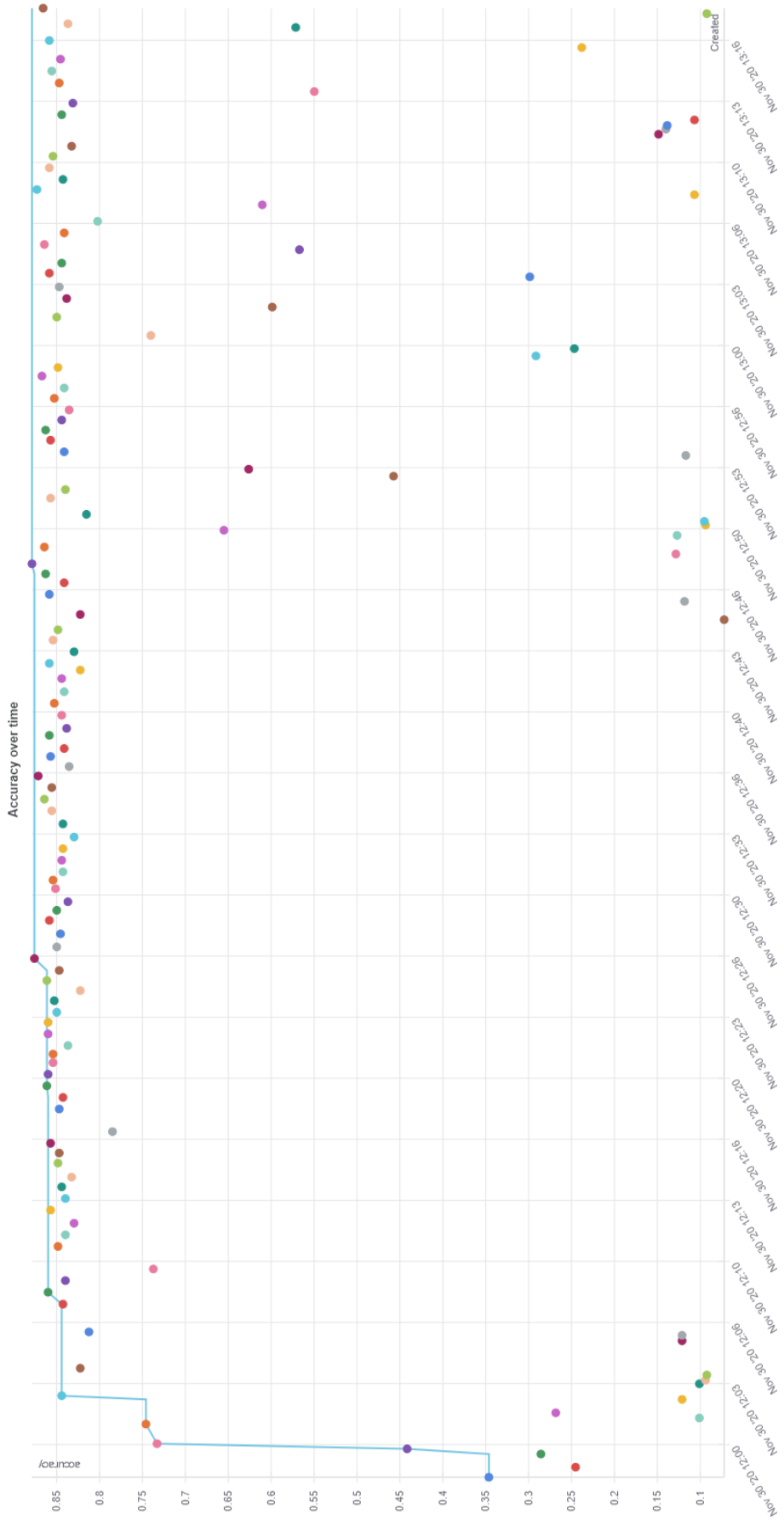

Figure 7: Accuracy over time - Multilayer Perceptron

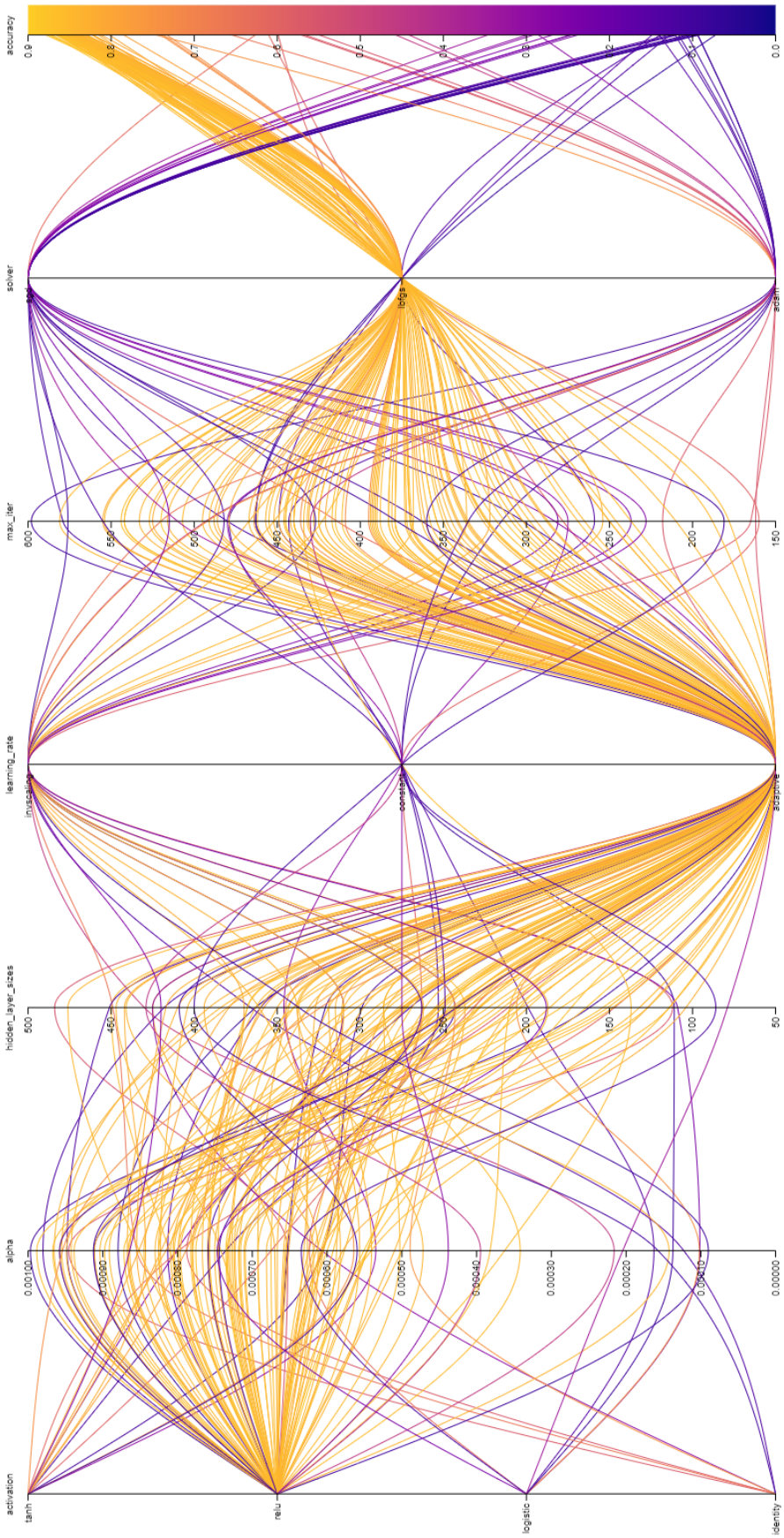

Figure 8: Multilayer Perceptron Parameters