Dissertation Defense: Automated Feature Engineering for Deep Neural Networks with Genetic Programming

Jeff Heaton

Nova Southeastern University - Ft. Lauderdale, FL USA

March 3, 2017

Dissertation Chair: James Cannady, Ph.D.

Dissertation Committee: Sumitra Mukherjee, Ph.D. & Paul Cerkez, Ph.D.

Abstract

Background

Feature engineering is a process that augments the feature vector of a machine learning model with calculated values that are designed to enhance the accuracy of a model's predictions. Research has shown that the accuracy of models such as deep neural networks, support vector machines, and tree/forest-based algorithms sometimes benefit from feature engineering. Expressions that combine one or more of the original features usually create these engineered features. The choice of the exact structure of an engineered feature is dependent on the type of machine learning model in use. Previous research demonstrated that various model families benefit from different types of engineered feature. Random forests, gradient-boosting machines, or other tree-based models might not see the same accuracy gain that an engineered feature allowed neural networks, generalized linear models, or other dot-product based models to achieve on the same data set.

Abstract (cont.)

Proposed Research

This dissertation presents a genetic programming-based algorithm that automatically engineers features that increase the accuracy of deep neural networks for some data sets. For a genetic programming algorithm to be effective, it must prioritize the search space and efficiently evaluate what it finds. This dissertation algorithm faced a potential search space composed of all possible mathematical combinations of the original feature vector. Five experiments were designed to guide the search process to efficiently evolve good engineered features. The result of this dissertation is an automated feature engineering (AFE) algorithm that is computationally efficient, even though a neural network is used to evaluate each candidate feature. This approach gave the algorithm a greater opportunity to specifically target deep neural networks in its search for engineered features that improve accuracy. Finally, a sixth experiment empirically demonstrated the degree to which this algorithm improved the accuracy of neural networks on data sets augmented by the algorithm's engineered features.

Outline I

1. Introduction

Introduction

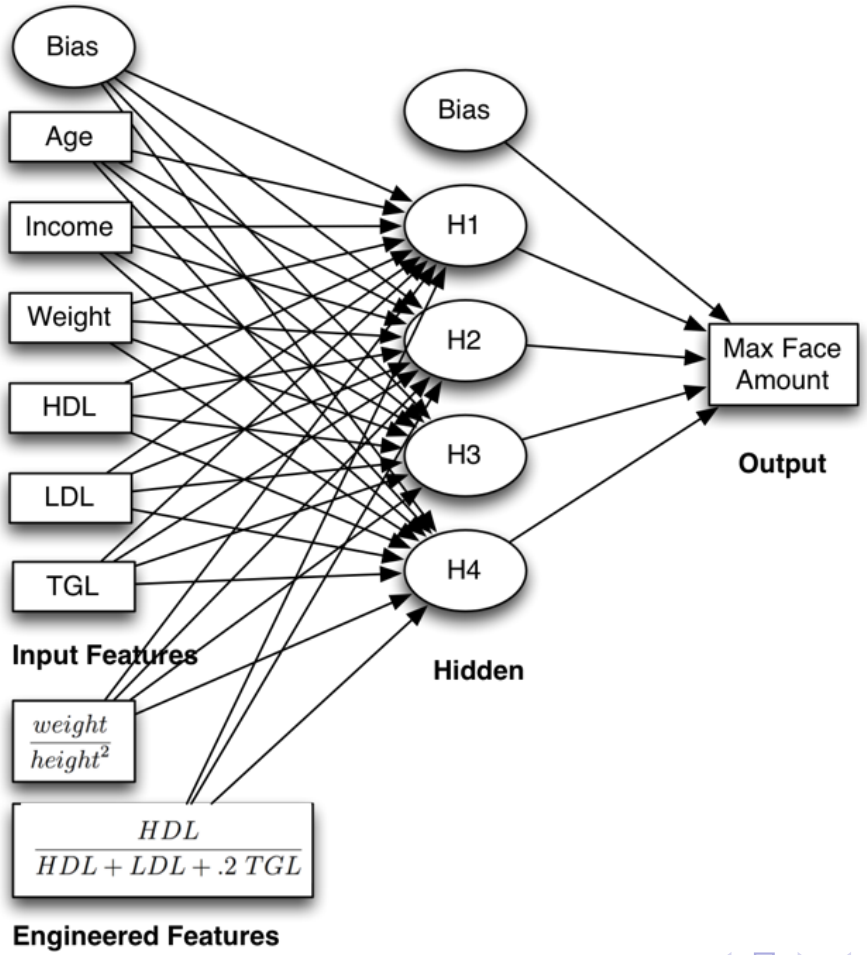
Sample Insurance Data

Age	Income	Weight	HDL	LDL	TGL	Amount	Class
20	73,300	183.9	36	114	209	281,814	p
74	10,216	243.1	64	138	188	0	d
15	23,180	213.0	45	159	89	0	d
21	50,764	134.8	45	189	103	206,770	p
31	59,358	127.8	63	163	94	232,696	p
67	96,923	183.3	42	100	155	340,382	s
31	96,996	159.8	32	139	123	351,549	s
22	38,581	225.3	39	123	131	0	d
37	22,984	197.1	58	136	185	87,496	p
60	50,283	182.2	40	104	124	184,937	s
76	89,534	225.8	58	137	124	0	d
49	85,201	184.8	69	111	92	312,734	s
30	96,818	140.5	66	98	110	378,208	p

Introduction

Neural Network & Feature Engineering Overview

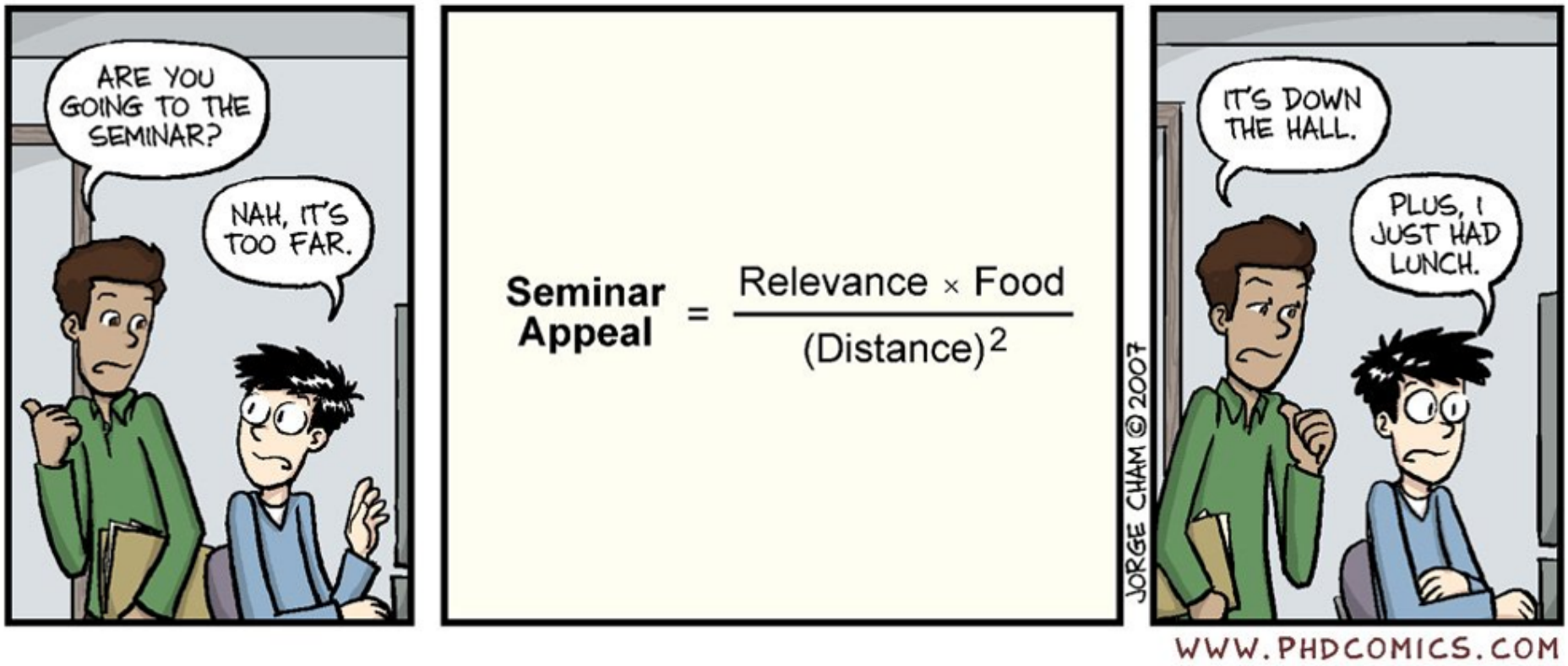
Feature engineering essentially creates new features based on expressions of the original input features.



Introduction

Engineered Features

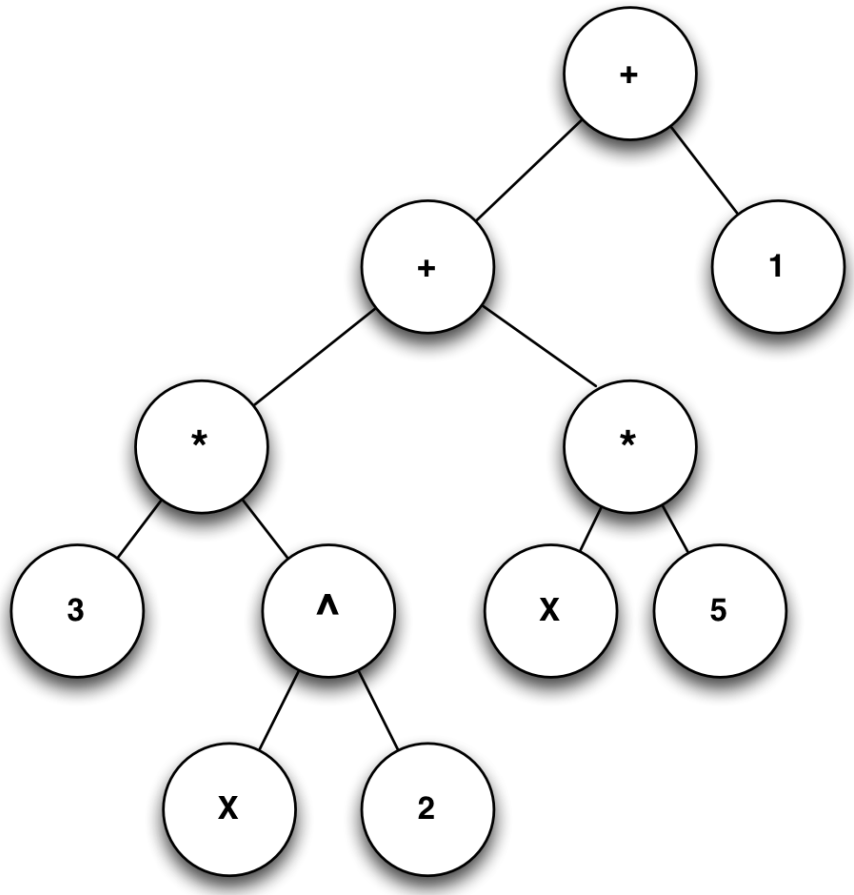
Often built from ratios, powers, products, etc.



Introduction

Genetic Programming Overview

Genetic programming evolves expressions. This dissertation used tree representations.



There is currently no automated means of engineering features specifically for a feedforward deep neural network that are a combination of multiple named features from the original feature vector.

- The proposed algorithm specifically targets features for feedforward deep neural networks. Convolutional neural networks (CNNs) and recurrent neural networks (e.g. LSTM) are not covered by this research.
- Named features represent individual values, such as height, weight, age, amounts, and other measures.
- Unnamed features, such as pixels, audio samples, or word frequencies, are not covered by this research.

Introduction

Dissertation Goal

- Create an algorithm that will analyze a dataset and automatically create engineered features that will benefit a deep neural network.
- Engineered features will consist of mathematical transformations of one or more of the original features from the dataset.
- Success is measured as the decrease in root mean square error (RMSE) when a neural network's feature vector is augmented with engineered features produced by the algorithm.
- Results will be reported on a variety of real-world and synthetic datasets.

Other/Prior Research by Jeff Heaton

Previous Publications

- Heaton, J., McElwee, S., Cannady, J., & Fraley, J. (May 2017). Early stabilizing feature importance for TensorFlow deep neural networks. In International *Joint Conference on Neural Networks (IJCNN 2017)* (accepted for publication). IEEE.
- Heaton, J. (2016, March). An empirical analysis of feature engineering for predictive modeling. In *SoutheastCon 2016* (pp. 1-6). IEEE.
- Heaton, J. (2015). Encog: Library of interchangeable machine learning models for Java and C#. *Journal of Machine Learning Research*, 16, 1243-1247.

Prior Research by Jeff Heaton

An Empirical Analysis of Feature Engineering for Predictive Modeling

Heaton (2016) investigated the ability of neural networks to learn the following features. Red entries had the most difficulty and are good candidates for feature engineering for neural networks.

- **Counts:** Count of 50 features above a threshold.
- **Differences:** $y = x_1 - x_2$
- **Logarithms:** $y = \ln(x_1)$
- **Polynomials:** $y = 1 + 5x + 8x^2$
- **Powers:** $y = x^2$
- **Ratios:** $y = \frac{x_1}{x_2}$
- **Rational Differences:** $y = \frac{x_1 - x_2}{x_3 - x_4}$
- **Rational Polynomials:** $y = \frac{1}{5x + 8x^2}$
- **Root Distance:** $y = \left| \frac{-b + \sqrt{b^2 - 4ac}}{2a} - \frac{-b - \sqrt{b^2 - 4ac}}{2a} \right|$
- **Square Roots:** $y = \sqrt{x}$

2. Automated Feature Engineering (AFE) Algorithm

AFE Algorithm

5 Experiments to Design an Algorithm

- **Experiment 1:** Limiting the Search Space
- **Experiment 2:** Establishing Baseline
- **Experiment 3:** Genetic Ensembles
- **Experiment 4:** Population Analysis
- **Experiment 5:** Objective Function Design

AFE Algorithm

Experiment 1-5 Results

- Larger data sets take considerable time for the GP to fit.
- Feature ranking is unstable until neural network training has finalized.
- GP is very prone to finding many local optima.
- GP produces many different candidate solutions with pronounced sub-patterns.
- Coefficients will vary greatly in solutions created by genetic algorithms.

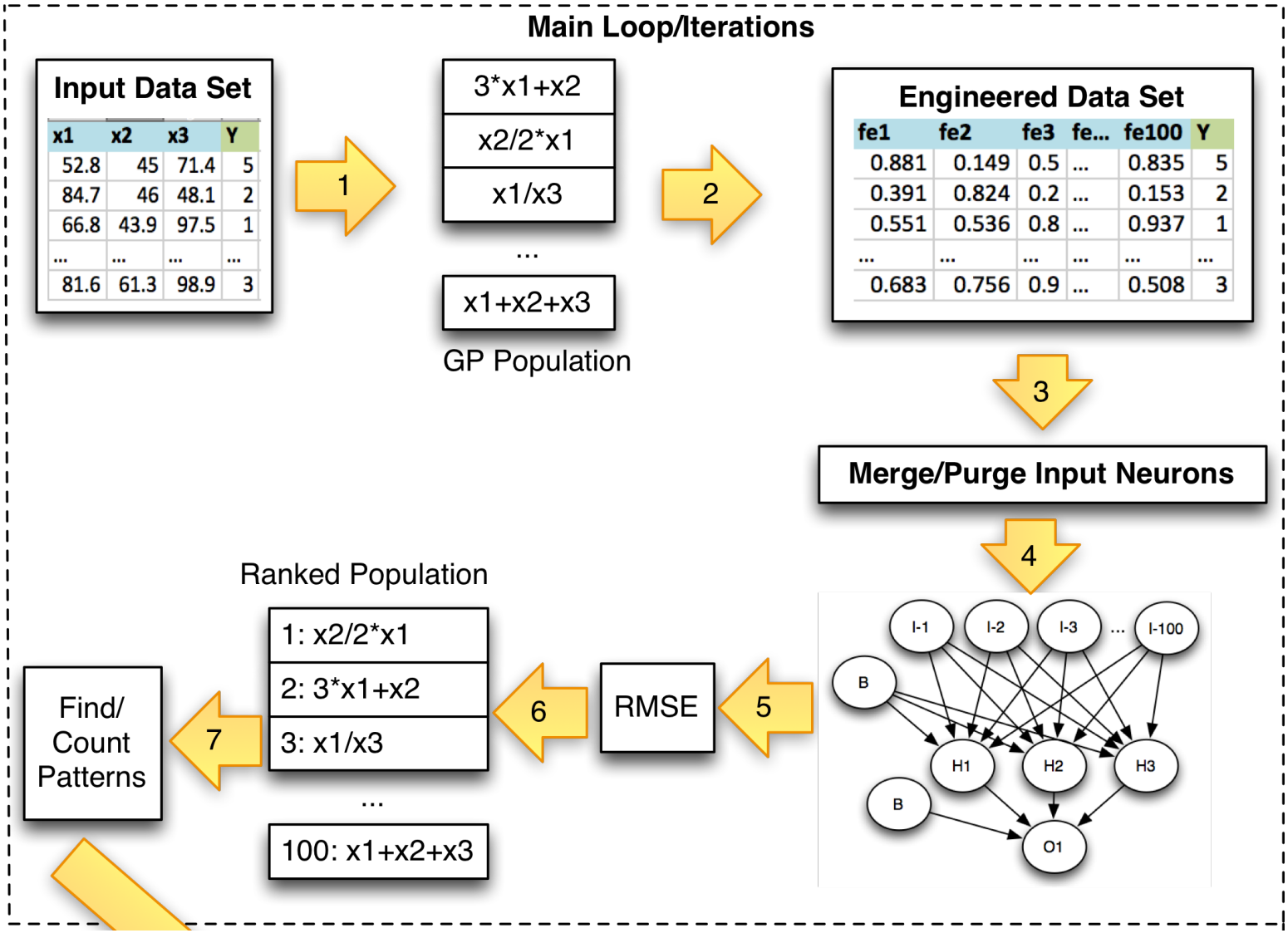
AFE Algorithm

Primary Components

- **Single Evolving Neural Network:** Place the entire population of engineered features into the neural network feature vector. Do not evaluate engineered features one-by-one.
- **Feature Importance Sorting:** Population members will not be scored. They will be sorted according to feature importance. The neural network is trained until this ranking stabilizes.
- **Use Common Patterns:** Build final engineered features by analyzing common patterns across the entire population.
- **Optimize Coefficients with Gradient Descent:** Coefficients of GP expressions can be optimized with gradient descent. These coefficients will be adjusted to values that minimize the error of a neural network where its data set is augmented with these features.

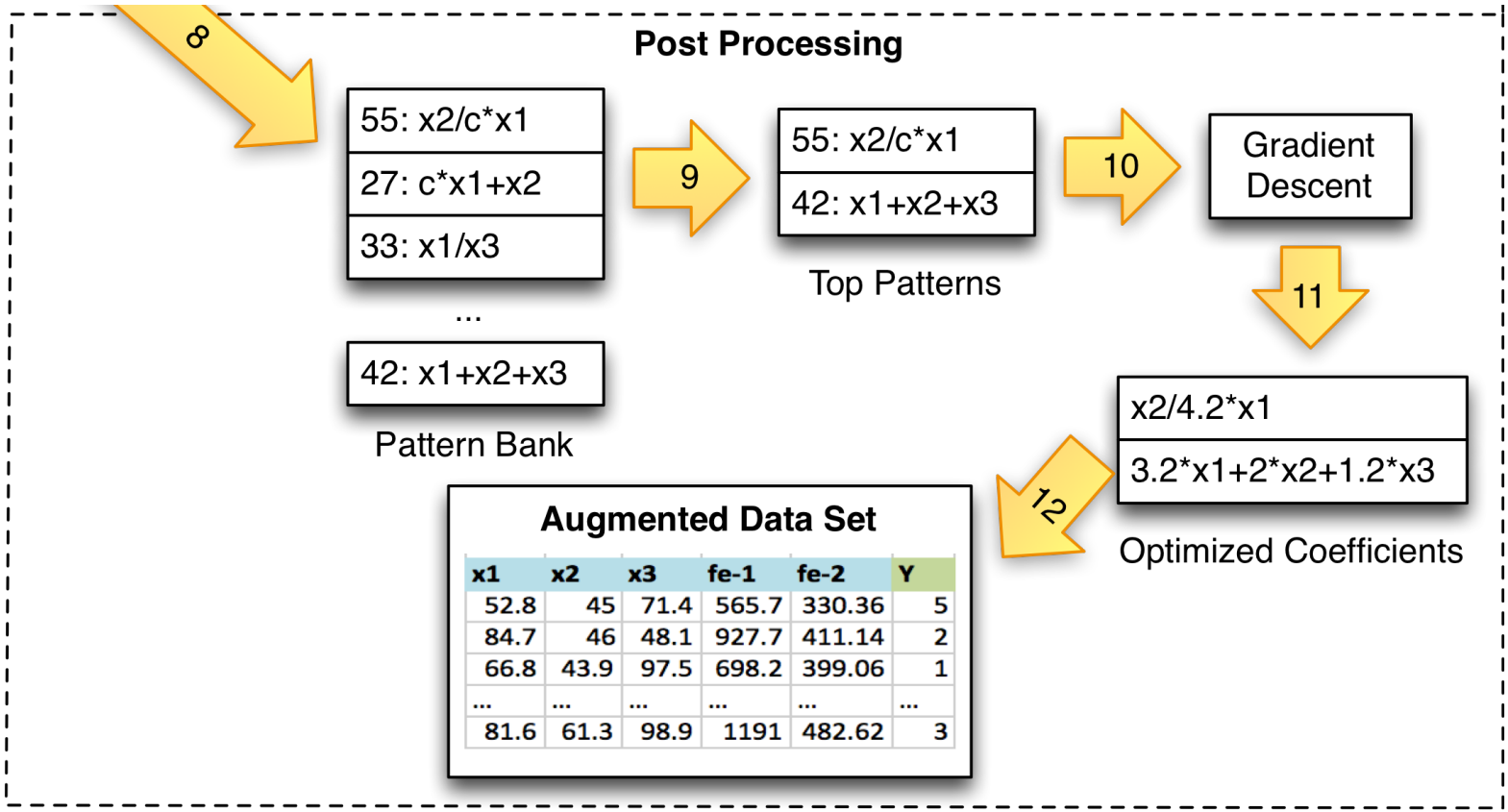
AFE Algorithm

High Level Overview - Iterations



AFE Algorithm

High Level Overview - Post Processing



3. Evaluating the AFE Algorithm

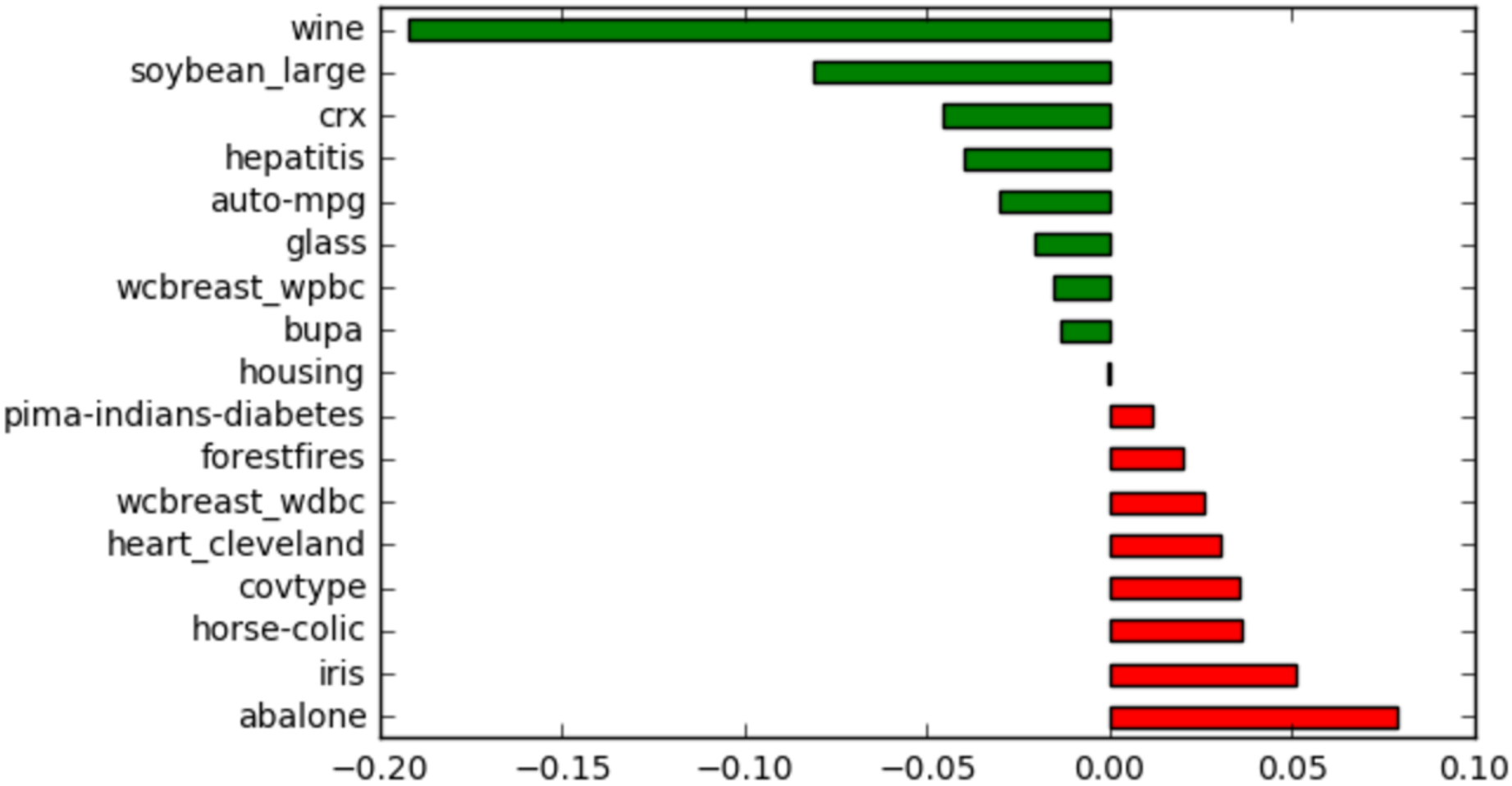
Evaluating the AFE Algorithm

Experiment 6 Data Set Result Table

#	Name	Neural Mean	AFE Mean	Difference	Percent Difference
6-1	Abalone	2.6523	2.8611	0.2087	0.07872
6-2	Auto-MPG	3.3376	3.2369	-0.100	-0.03016
6-3	Bupa	0.4551	0.4490	-0.006	-0.01357
6-4	Covtype	0.3999	0.4140	0.0141	0.035392
6-5	CRX	0.3592	0.3428	-0.0164	-0.04575
6-6	Forestfires	0.0607	0.0619	0.0011	0.019726
6-7	Glass	0.3376	0.3306	-0.007	-0.02078
6-8	Heart	0.3377	0.3480	0.0102	0.03038
6-9	Hepatitis	0.4576	0.4392	-0.018	-0.0401
6-10	Horse	0.3659	0.3792	0.0133	0.0363
6-11	Housing	0.0446	0.0446	0	-0.0009
6-12	Iris	0.3182	0.3345	0.0162	0.05118
6-13	Pima	0.4163	0.4212	0.0048	0.01176
6-14	Soybean_large	0.1936	0.1779	-0.0157	-0.0811
6-15	WCbreast_wdbc	0.0988	0.1014	0.0025	0.02587
6-16	WCbreast_wpbc	0.3811	0.3752	-0.0059	-0.0156
6-17	Wine	0.2520	0.2036	-0.0484	-0.1922

Evaluating the AFE Algorithm

Experiment 6 Data Set Result Chart



4. Fulfilling the Dissertation Goal

Fulfilling the Dissertation Goal

Statistical Significance of the Results

#	Name	T-Statistic	P-Value
6-2	Auto-MPG	0.38309	0.702064
6-3	Bupa	1.190632	0.235223
6-5	**CRX**	**4.115739**	**0**
6-7	Glass	1.857998	0.064654
6-9	**Hepatitis**	**6.679419**	**0**
6-11	Housing	0.088687	0.92942
6-14	Soybean_large	3.038647	0.002697
6-16	WCbreast	1.125605	0.261695
6-17	**Wine**	**3.030937**	**0.002764**

Fulfilling the Dissertation Goal

Engineered Feature from the Wine Data Set

$$\frac{(7.54^{tp^{2.96}} - ah + \frac{10alh}{9} + mg - 6)^{\frac{ah}{h}}}{1 + 0.06mg - 0.01p}$$

- *a* - alcohol
- *ah* - ash
- *alh* - alcalinity_ash
- *ci* - color_intensity
- *f* - flavanoids
- *h* - hue
- *ma* - malic_acid
- *mg* - magnesium
- *nfp* - nonflavanoid_phenols
- *od* - od28_od315
- *p* - proanthocyanins
- *pr* - proline
- *tp* - total_phenols

5. Future Work

Future Work

Research Direction

- Replace Encog with TensorFlow and DEAP.
- Include SymPy for differentiation and simplification.
- Grid processing/scaleability.
- Expand data set evaluation to include Kaggle data sets.
- Try/improve algorithm on a live Kaggle competition.

Future Work

Publication Goals

My plan is to extend and publish research from this dissertation in either a journal or a conference proceeding. The following are possible venues, in order of preference:

- Journals:
  - **Journal of Machine Learning Research (JMLR)** (ISSN 1532-4435), `http://www.jmlr.org/`
  - **Genetic Programming and Evolvable Machines** (ISSN: 1389-2576), `http://www.springer.com/computer/ai/journal/10710`
- Conferences:
  - **Genetic and Evolutionary Computation Conference (GECCO)**, `http://sig.sigevo.org/`
  - **International Joint Conference on Neural Networks (IJCNN)**, `http://www.ijcnn.org/`