Bilateral Trade Flow Prediction

Sonia Circlaeys, Chaitanya Kanitkar, Daiki Kumazawa

***Abstract*—In this paper, we seek to improve prediction of bilateral trade flow, an important economic indicator used by economists and policy makers. The Gravity Model of Trade, a linear regression-based model, is a commonly used empirical method which we use as our baseline. Using a fully-connected, feedforward neural network, we were able to successfully improve upon Gravity Model's prediction accuracy by an R^2 score of .15 measured on CEPII's "Tradhist" data set. This is significant because our findings show that machine learning methods are effective in prediction of economic variables using only time-agnostic features. We present our findings here and discuss future opportunities for further investigation.**

I. Introduction

Bilateral trade flow is an important economic indicator used by economists and policy makers. It represents the value of goods and services that have been exported from one country to another, influencing international trade policy as well as domestic economic policy in both countries. For example, an increase in exports from China to Venezuela would exacerbate Venezuela's trade balance (i.e. exports minus imports.) As a result, Venezuela may need to fill the financial shortfall created by this increased outflow of money to China, while China will likely benefit from the influx of foreign assets from Venezuela. Since these deficits/surpluses caused by international trade are considered a major factor in the process of a country's economic development,[1] studying bilateral trade flow is an important research agenda.

This project applies Machine Learning techniques to achieve superior prediction of bilateral trade flow. The input to our algorithms is a set of economic and geographic variables, such as GDP and the distance between the importer and exporter country. Our input space will be discussed extensively in the feature section. We use a linear regression, with raw features as well as logarithmic features, a kernelized linear regression, and a neural network with a variety of architectures. The output of these algorithms is bilateral trade flow value measured in British Pound Sterling (GBP). We have found that neural networks are a promising approach due to their capacity to capture nonlinear interactions between features.

This paper is organized as follows. In the next section, we will provide an overview of existing literature and discuss our work's contribution. The third section introduces the data set and cleansing techniques used, and provides a detailed account on features. The fourth section provides discussions on our model selection. The fifth and sixth sections present and discuss the performances of our proposed models. The seventh section concludes.

[1] See for example [1]

II. Related Work and Contribution

There is extensive economics literature that studies bilateral trade flow. The traditional approach has been to use an empirical method called the Gravity Model of Trade. Motivated by Newton's law of universal gravitation, Tinbergen [2] proposed to model the bilateral trade flows between two countries using GDPs of the origin and destination countries as well as the distance between the two countries:

$$FLOW = \alpha \frac{GDP_o^{\beta_1} GDP_d^{\beta_2}}{Dist^{\beta_3}} \tag{1}$$

Ever since, the model has been used extensively because of its empirical power, although the model is controversial among economists as it lacks rigorous theoretical justification.

Most of economics literature has focused mainly on theoretically justifying, or sometimes disproving, the gravity model in order to make causal inferences about the bilateral trade amount (see, for example, [3] for a summary of literature on Gravity Model.) Making theoretically justified, correctly specified models has been an important goal for economists and policy makers because misspecified models will lead to erroneous statistical interpretations of independent variables' parameters. Anderson [4] and Bergstrand [5] are some of the earliest attempts to corroborate the Gravity Model by incorporating international trade theories into the intuitive application of gravitational law.

Rather than concerning ourselves with making theoretically correct models for causal inference, our project instead attempts to attain superior predictive ability of bilateral trade flow using supervised machine learning methods.

In this sense, our approach is similar to a branch of econometrics that is focused more on forecasting rather than on inference, namely time series econometrics. Thus, we consider our work to follow the spirit of time series models, such as AR, MA, ARMA type models. Lacking rigorous theoretical justification, our models will not be useful in studying the underlying mechanisms of bilateral trade. Nonetheless, we believe this is an important research objective since the amount of exports influences governments' domestic and trade policies, and a model that provides more powerful trade volume prediction would be of good use to policy makers.

While there is abundant existing literature on Gravity Models, applying machine learning methods to predict trade flow remains a new research topic. One example is Nummmelin and Hanninen [6] that used the Support Vector Machine to analyze and forecast bilateral trade flows of soft sawnwood. More relevant to our objective, Nuroglu [7] used data for 15 EU countries and showed that neural networks achieve a lower MSE compared to panel models. Therefore, our project is one of the earliest attempts that use a variety of methodologies

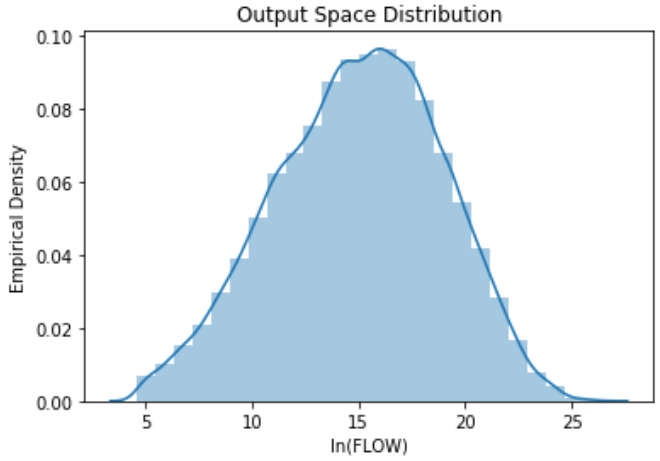

Fig. 1. Histogram of the log of trade flow after cleaning

in machine learning to tackle this research question, using a larger data set from CEPII that includes 200+ countries.

III. DATA SET AND FEATURES

A. Data Set

Our primary data source is the bilateral trade and economic variable data set "Tradhist" distributed by CEPII [8]. The data set contains 1.9 million data points from the 1800s to 2014 for 200+ different countries and territories. Our data set has both geographical dimension (country-pairs, such as USA to China, France to Germany, etc.) and time dimension.

B. Cleansing Techniques Employed

To avoid abnormalities in data and potential obstacles in the process of prediction, we have applied several data cleansing techniques before we begin our analysis. First, we removed incomplete data from the data set. That is, we did not work with data points that have missing feature values. Second, we removed all trade flow values that are below 100 GBP, as they are not economically significant and cause out-lier problems. Third, we took the log of data to achieve a smoother distribution of the data (See Figure 1). Finally, we removed data before 2009 as our focus is to predict recent trade flows.

C. Features

Table I summarizes the 12 economic features that are used in our models. GDP and the amount of exports are reported in current GBP.

Most of these variables are commonly used in the framework of the Gravity Model and thus are well justified. Making prediction using these features enables us to compare our performance against the Gravity Model, which is our baseline model.[2]

[2]We note that one alternative approach in feature selection is to collect more features and use selection procedures such as forward search to pick features that produce better generalization errors. However, since we would like to compare our proposed models' performances against the baseline Gravity Model, and since Gravity Model have traditionally used features that we listed above, we decided to use a similar set of features as Gravity Model does.

TABLE I
FEATURES

GDP_o	GDP of origin country
GDP_d	GDP of destination country
Pop_o	Population of origin country
Pop_d	Population of destination country
Dist	'Great circle distance' between two countries
Comlang	1 if at least one language is spoken by more than 9% of the population of both countries
Contig	1 if two countries are contiguous
OECD_o	1 if the origin country is an OECD member
OECD_d	1 if the destination country is an OECD member
GATT_o	1 if the origin country is a GATT member
GATT_d	1 if the destination country is a GATT member
XPTOT_o	Total value of exports of the origin country

We also note that for continuous variables such as GDP, distance, and population, we tried both the log version and non-log version in our feature set. The log version was used in most of our training models for the following reasons:

1) In Gravity Model, taking the log of equation (1) will make the function linear in parameters, enabling us to run a linear regression.
2) As we have discussed in the data cleansing section, taking the log achieves a smoother distribution of data, thereby avoiding issues that stem from scaling of values.

IV. METHODS

We have selected six Machine Learning models for our project. In this section, we will discuss how we chose these models and briefly describe what the proposed models are. Before proceeding to presenting the models we consider, we note that we mostly focus on making predictions of the current trade flow using time-agnostic features, except for the last two models in which we explicitly include the past lagged values of trade flow to the feature set. For example, in predicting the trade amount for time t, we use GDPs of both origin and destination countries of time t, and not the past values from time $t-1$, $t-2$, etc. Likewise, all other features that are used in prediction are of the same time period. An obvious alternative approach would be to include the past values, but up to the fourth model that we present later, we attempt to achieve a generalizable, time-agnostic model that can better predict trade flow.

We outline some of the considerations we made in selecting our models. First, given that our objective is to predict a continuous variable, we have focused on regression-type algorithms rather than on classification ones. Of all options available, we excluded generative learning algorithms because we have no theoretical or empirical evidence in making assumptions on each feature's conditional probability distribution. Therefore, methods such as Naive Bayes, Gaussian Discriminant Analysis and the like are not fit to our problem.

We further presumed that models that capture interactions among features may provide a better prediction performance. For this reason, we implemented the linear regression with different kernels. Using kernels will likely capture any relations beyond ones that are linear, thereby potentially improving predictive performance. Likewise, we have also opted for neural

networks because of their robustness in non-linear settings. Among the family of available neural network methods, we have focused on fully-connected, feedforward methods. This is due to the fact that since our data has both geographical and time dimensions, setting up recurrent neural networks would pose a challenge and thus is discussed as an option for future investigation.

Therefore, our objective is to implement and compare performances of the following methods:

1. Ordinary Linear Regression with Regularization

This is our baseline model, where we regress the raw amount of bilateral trade flow on the set of features:

$$FLOW = X\beta + \epsilon \tag{2}$$

L2 regularization will be included to avoid potential overfit issues.

2. Gravity Model (Linear Regression with logarithmic features)

As discussed above, Gravity Model describes the interactions among the GDPs and distance in a functional form inspired by Newton's law of universal gravitation. Taking the log of equation (1), we obtain

$$\begin{aligned} ln(FLOW) = \alpha' + \beta_1 ln(GDP_o) \\ +\beta_2 ln(GDP_d) + \beta_3 ln(Dist) \end{aligned} \tag{3}$$

which is linear in parameters. To this baseline specification, we will add other economic features of interest discussed above.

3. Kernel Ridge Regression (RBF and Polynomial Kernel) with Logarithmic Features

Departing from linear models, we investigate whether kernels would successfully improve prediction by capturing non-linearity. In particular, we employ the RBF kernel

$$K(x,z) = exp(-\frac{||x-z||^2}{2\sigma^2}) \tag{4}$$

and the polynomial kernel

$$K(x,z) = (x^T z + c)^2 \tag{5}$$

4. Fully Connected, Feedforward Neural Networks using Logarithmic Features

Since we assume no a priori knowledge on interactions between features (except for Gravity Model), we utilized a fully connected neural network. We note that this was computationally acceptable because our feature space was relatively small (12 features). After experiments, we decided to use logarithmic features, since this achieves a smoother distribution of data as we discussed in the data cleansing and thus leads to stronger predictive abilities.

To find an optimal architecture of the neural network, we will compare architectures with different layer and node setups.

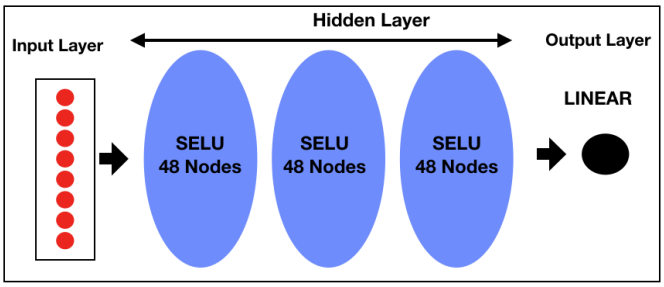

Fig. 2. Fully connected, feedforward neural network

5. Linear Regression with Autoregressive Model (AR(1))

Departing from the models that use features from the same time period, we also investigate the performance of time series models against the other proposed models. We use a traditional time series model, where we regress the future value of trade flow on the past value:

$$ln(FLOW_t) = \alpha + \beta ln(FLOW_{t-1}) + \epsilon \tag{6}$$

6. Fully Connected, Feedforward Neural Networks using Logarithmic Features and Lagged Output Values

Here, we add the lagged values of the trade flow to the original feature space. This can be considered as a natural extension of the Gravity Model and pure time series models.

V. Results

We formally present the results of our models in this section.

A. Comparison Metrics and Validation Method

To compare the predictive performances of our models, we used R^2, the coefficient of determination, as our metric:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} = 1 - \frac{\sum_i (\hat{y}_i - \bar{y})^2}{\sum_i (y_i - \bar{y})^2} \tag{7}$$

This metric approaches one when our predicted values are close to true values, and is commonly used as a measure for goodness of fit.

As a validation method, we conducted a hold-out validation using 30% of the entire data set. Since we had enough data, we simply separated the data set into a training set and a test set instead of conducting k-fold validation. The original data set is separated into the train set of 91,747 examples (70 %) and test set of 39,321 examples (30%). We note that for the kernelized linear regression, we have constrained our analysis to a subset of the training set due to computational issue.[3]

B. Hyperparamter Tuning

1) Regularizer: For our linear regression based models, we used L2 regularizer to counteract overfitting of our model. Because of an overabundance of data, we were able to use dev sets to decrease model overfitting. For our neural network based models, we fine-tuned kernel regularizer (weight regularizer) per hidden layer.

[3]We will provide detailed discussion on the computational challenge in the following section.

TABLE II
MODEL PERFORMANCES

Models	Train Set MSE	R^2	Test Set MSE	R^2
1	1.4×10^{19}	0.14	1.4×10^{19}	0.13
2	6.45	0.64	6.53	0.63
3	5.07	0.72	8.38	0.61
4	2.18	0.88	3.77	0.79
5	1.32	0.89	1.31	0.89
6	1.57	0.90	1.59	0.90

2) Number of Hiddnen Units/ Layers: We experimented with different dense network architectures by manipulating the number of hidden nodes and layers for our network. We will discuss this point in detail in the next section.

3) Batch Size: We used the Keras neural network API built on top of TensorFlow to train and test our networks, and therefore our batch size was also dependent on the implementation of the API. To balance the computational time required to loop through our batches, computational time of computing large batch gradients, as well as the time required for convergence, we found an optimal batch size of 1,000 data points that minimized our cost function the most for a given amount of time. Also we found that batch sizes smaller than 1,000 started to oscillate at later epochs and therefore struggled to converge.

4) Learning rate and Optimizer: We discovered that adaptive learning rate optimizers were most effective at achieving faster convergence. We measured performances of different optimizers for our problem, which are reported in Figure 6. This point will also be discussed further in the next section.

C. Feature Analysis

Figure 4 reports the top six features with highest coefficients in linear regression. The results confirm Gravity Model's empirical success in that features such as GDPs and distance are dominant factors influencing the trade amount. This analysis also showed that sharing a common language and being part of World Trade Organizations is also influential in predicting trade flow.

D. Findings

Table II presents the final performances of the six models. The table reports both the mean squared error as well as the R^2 value of each model. Table III summarizes the performances of different neural network architectures. The table reports R^2 on the test set after 200 epochs for each of the architectures examined. The depth denotes the number of hidden layers, and the width denotes the number of nodes in each hidden layer. Figure 3 provides a typical plot of models' performances against the ground truth. The figure shows both predicted and actual amount of exports of Nigeria in 2012. The x-axis represents the destination country, and the y-axis is the logarithmic value of trade amount.

TABLE III
TEST SET R^2 SCORES FOR DIFFERENT NEURAL NETWORK ARCHITECTURES (AFTER 200 EPOCHS)

		Depth		
		1	3	5
Width	6	.661	.663	.664
	12	.677	.696	.701
	48	.689	.765	.780

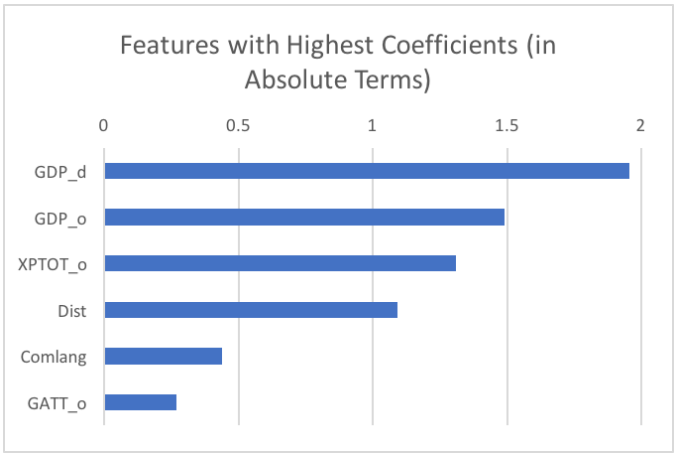

Fig. 4. Top 6 features with highest coefficients

VI. OBSERVATION AND DISCUSSION

First, kernelized regression was not a viable method in practice. Despite its potential to work well with non-linearity, given the large size of our data set, there were too many parameters to train, posing a computational challenge.

Second, as Table III shows, wider neural networks were more successful than narrower ones for our problem. We initially hypothesized that non-linear models may be a successful approach. Confirming that hypothesis, casting the input layer into a higher dimensional space seems to have done better than those that used less number of nodes in hidden layers. After careful validations, we have reached the optimal architecture of 3 hidden layers with 48 nodes each. Deeper network architectures fit the training set better, but did not achieve any superior predictive ability on the test set.

Third, fully connected, feedforward neural network with logarithmic features (Model 4) achieved a remarkably better predictive performance compared to Gravity Model (Model 2). As Table II shows, using the same set of features as Gravity Model, Model 4 achieved a remarkable improvement of above .15 in the test set's R^2 score. This seems to indicate that neural networks are successful at discovering non-linear interactions between features compared to the Gravity Model, which is a purely linear model of logarithmic features. This demonstrates that we were able to achieve superior predictive ability without resorting to time series models, which was one of the goals of our project.

Fourth, optimization issues with the neural network are overcome by selecting an effective activation function and optimizer. Figure 5 shows that the hyperbolic tangent attains the best performances among other alternative activation

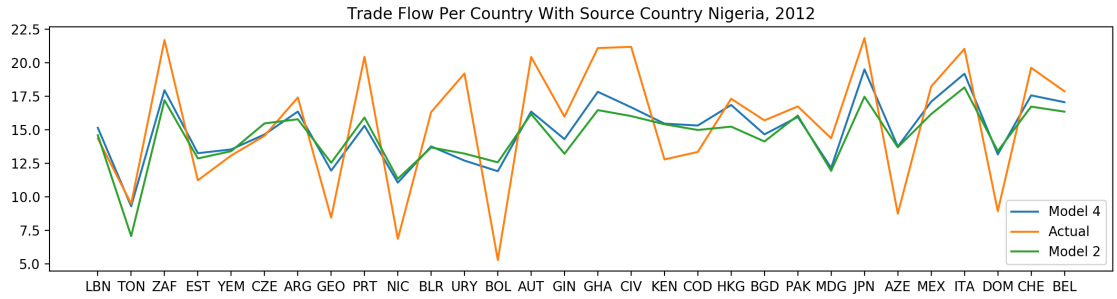

Fig. 3. Typical plot of model performances against actual trade amount

functions.[4] We observe that activation functions with stronger gradients generally perform better than sigmoid, confirming the known issue of vanishing gradient. Furthermore, as Figure 6 shows, optimizers that use adaptive learning rates seem to perform better than the stochastic gradient descent.

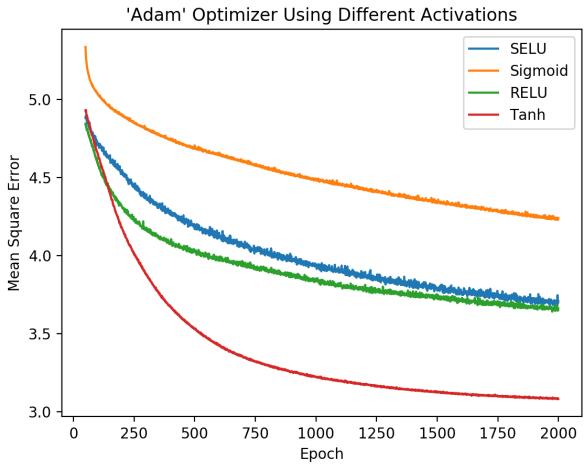

Fig. 5. Mean squared error plotted over 2000 epochs: different activation functions

Finally, Model 6 achieved the best performance among all models we investigated. The model requires knowledge of past trade flow, but it produces the most accurate results with R^2 above .9. This is expected as this model is a combination of the AR model, gravity model, and neural networks. We also note that this model was only able to improve Model 5 by a small amount, likely due to the fact that there is not much non-linearity that can be exploited when we use the past value, which is by itself a good prediction of future values.

VII. Conclusion

Our results have shown that using neural networks is a promising approach in predicting bilateral trade flow when we are making predictions with other economic variables of the same time period. Specifically, fully-connected, feedforward

[4]As per common practice to keep our network simple, we used the same activation function across all layers in the neural network except the last one.

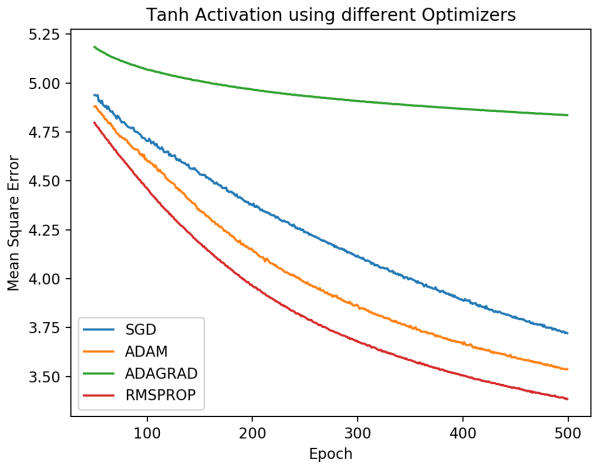

Fig. 6. Mean squared error plotted over 500 epochs: different optimizers

neural network was able to improve Gravity Model's prediction performance by the test set R^2 score of .15, using the same set of features.

On the other hand, neural networks do not seem to significantly outperform AR, a traditional time series econometric model. This might be due to the fact that there is not much nonlinearity that can be exploited in the data when we use the past lagged values.

We propose several avenues for future research. Since our data has both geographical and time dimensions, using the LSTM model extended to a panel setting (for example, a model developed in [9]) may be a fruitful approach that we can take in the future. Discretizing our output space rather than using a continuous output space may also produce more accurate results. Since the neural network approach showed promise with capturing nonlinear interaction effects among other economic indicators, including many more economic features may also be effective in further improving our prediction.

VIII. Tools Used

1) Pandas for data management [10]
2) Scikit-learn for linear regression/ kernels [11]
3) Matplotlib for plotting [12]
4) Keras for neural networks [13]

IX. Contributions

We contributed equally at all phases of the project, from feature selection, model implementation, to preparation of deliverables. Because of his economics background, Daiki took a larger role in data collection, manipulation and literature review. Chaitanya utilized his software development expertise to polish and tune our models and algorithms.

References

[1] M. P. Todaro and S. C. Smith, *Economic development, 12th Edition.* Harlow, England: Pearson Education, 2015.

[2] J. Tinbergen, *Shaping the World Economy: Suggestions for an International Economic Policy.* New York: Twentieth Century Fund, 1962.

[3] J. E. Anderson, "The Gravity Model," *Annual Review of Economics,* vol.3, pp.133-160, 2011.

[4] J. E. Anderson, "A theoretical foundation for the gravity equation," *American Economic Review,* vol.69(1), pp. 106-116, 1979.

[5] J. H. Bergstrand, "The gravity equation in international trade: some microeconomic foundations and empirical evidence," *The Review of Economics and Statistics,* vol.67(3), pp. 474-481, 1985.

[6] T. Nummelin and R. Hanninen, "Model for international trade of sawnwood using machine learning models," *Natural Resources and Bioeconomy Studies,* vol.74, 2016.

[7] E. Nuroglu, "Estimating and forecasting trade flows by panel data analysis and neural networks," *ktisat Fakltesi Mecmuas,* vol.64, pp.85-112, 2014.

[8] M. Fouquin and J. Hugot, "Two centuries of bilateral trade and gravity data: 1827-2014." *CEPII Working Paper,* 2016.

[9] M. Bai, B. Zhang and J. Gao, "Tensorial Recurrent Neural Networks for Longitudinal Data Analysis", https://arxiv.org/pdf/1708.00185.pdf, 2017.

[10] W. McKinney, "Data Structures for Statistical Computing in Python," *Proceedings of the 9th Python in Science Conference,* pp. 51-56, 2010.

[11] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research,* vol. 12, pp. 2825-2830, 2011.

[12] J.D. Hunter, "Matplotlib: A 2D graphics environment," *Computing In Science & Engineering,* vol.9(3), pp.90-95, 2007.

[13] F. Chollet *et al.*, Keras, Github, https://github.com/fchollet/keras, 2015.