Predicting Unknown Linguistic Behaviour from Known Linguistic Behaviour

L. L. Blumire

Abstract— A neural network is to be trained on data from WALS-3A[1], WALS-51A[2], WALS-83A[3], WALS-86A[4], and WALS-87A[5]. It will be given an input model, that allows for optional inputs of the features outlined in WALS, and an output model, that outputs its predictions for features. This will serve as an attempt to reveal underlying linguistic feature correspondences between morphological, syntactic, and phonetic information.

1 Introduction

Artificial Neural Networks (ANNs) facilitate the computation of large volumes of data that are not trivially decidable by an algorithm. They do this by simulating biological neural networks by creating a network of interconnected weights that are trained ('taught') based on output accuracy. In essence, this is done through a large amount of linear algebra.

They excel at forming classification models, and are also effective prediction models. The goal of this application is to attempt to predict linguistic properties ('features') of a language based on other known features.

This is to be done by collecting data on a set of WALS features. Only feature variants that are being considered are described in Section 2, for example English has no dominant genitive-noun order, and so will be excluded from the test data sets.

The hope is that by providing the options of features, as well as an unknown, the network can be trained to correctly predict a plausible output for it's network.

2 User Application

The goal of the application is to correctly model and predict linguistic features based on other linguistic features. This will hopefully reveal some underlying correspondences between syntactic, morphological, and phonetic information in natural human language.

The features outlined in Sections 2.1, 2.2, 2.3, 2.4, and 2.5 following are the features being used for the prediction model.

2.1 Consonant-Vowel Ratio

Consonant-Vowel Ratio represents the ratio between number of consonants and number of vowels in a language. It can be either Low, High, or Average. Thus it will represent 4 mutually exclusive input neurons, 'LowCVR', 'HighCVR', 'AverageCVR', and 'UnknownCVR'. For example, English has a Low ratio, meaning it has a high number of VowelQualities relative to it's number of consonants, so it would have 'LowCVR' consonants [1][6].

2.2 Position of Case Affixes

Position of Case Affixes represents the positioning of Case Affixes. They can be either suffixes, prefixes, or non-existent. Thus it will represent 4 mutually exclusive input neurons, 'PrefixCase', 'SuffixCase', 'NoCase', 'UnknownCase'. For example, English does not have a case system, so would be 'NoCase' [2].

2.3 Order of Object and Verb

Order of Object and Verb represents the positioning of objects relative to the verbs of a sentence. It can be either object first, or verb first. Thus it will represent 3 mutually exclusive input neurons, 'ObjectVerbOOV', 'VerbObjectOOV', 'UnknownOOV'. For example, English places object after the verb so would be 'VerbObjectOOV' [3].

2.4 Order of Genitive and Noun

Order of Genitive and Noun represents the positioning of possessors relative to their possessed object in a sentence. It can either be genitive first, or noun first. Thus it represents 3 mutually exclusive input neurons, 'GentitiveNounOGN', 'NounGenitiveOGN', 'UnknownOGN'. For example, French places the genitive after the noun, and so would be 'NounGenitiveOGN' [4].

2.5 Order of Adjective and Noun

Order of Adjective and Noun represents the positioning of descriptive terms and nouns in a sentence. It can be either adjcetive first, or noun first. Thus it represents 3 mutually exclusive input neurons, 'AdjectiveNounOAN', 'NounAdctiveOAN', 'UnknownOAN'. For example, English places the noun after it's describing adjectives, thus it would be 'AdjectiveNounOAN' [5].

2.6 Data Processing

After initial Data Processing, the number of languages with all 5 properties described on WALS is 247. This should provide a large enough sample size to train the neural network. It needs to be trained once for each property set as an unknown value looking to be discovered, and so therefore there would be six times as many entries (one set to train one to one, and one more for each of the unknown options). By taking two thirds of that data set giving 167 entries for training (1002 total) and 20 entries for validation (120 total) and 60 unseen entries for testing (360 total).

The code used to perform data processing is uploaded to github[7]

2.7 Network Initialisation

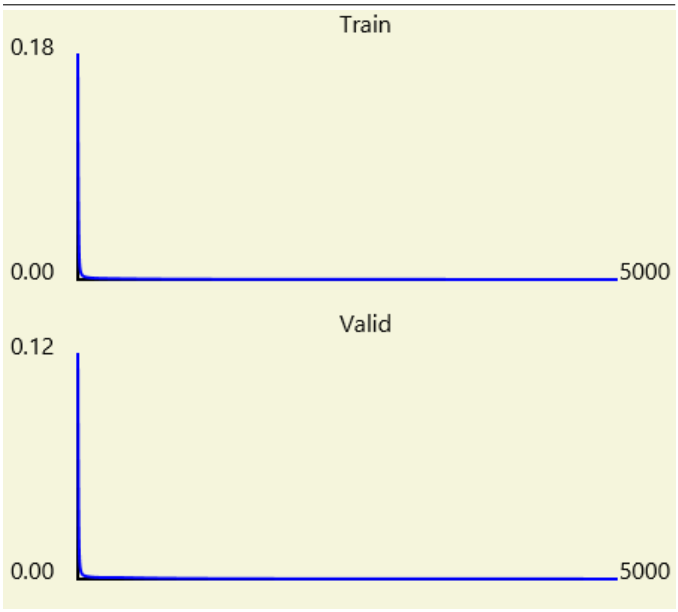
Each training unit should be evaluated equally, as that data has been processed procedural. As such, a learning momentum of 0 should be used. The program has a large amount of data to process, with a complex relationship underlying (if there is a relationship to be found), and thus should have a low learning rate of 0.01. There are 17 input neurons, and 12 output neurons—common advice would therefore say the average should be used as the number of hidden neurons, in this case 15.

3 Results of Application

The data was split into appropriate categories as discussed in Section 2.6. The data was then initialised, presented, and training was invoked.

The SSEs quickly found there way towards some incredibly low values, continuing to not increase, even after 5000 epochs.

Fig. 1 SSE Plots



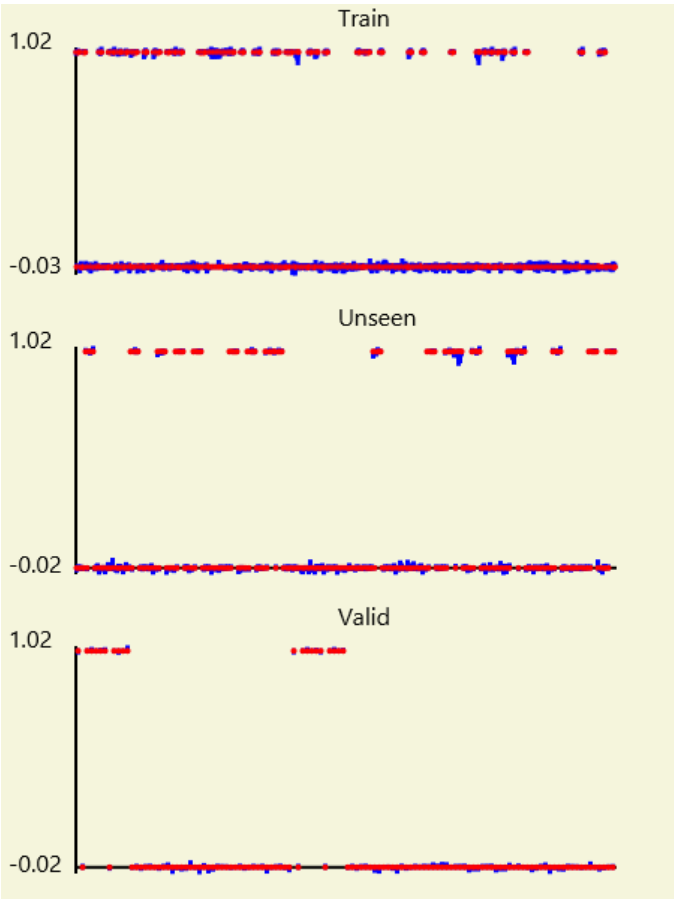

```
MLP - Users Train Valid and Unseen
15 Hidden Neurons Learn Rate 0.01
Momentum 0.00 Seed 12345
Train: SSE 0.2939 0.2362 0.3909 0.0224
0.1896 0.3352 0.1376 0.2004 0.3305
0.3011 0.1868 0.3089 Unseen: SSE
0.2970 0.2192 0.3946 0.0351 0.1953
0.3489 0.1421 0.1954 0.3254 0.2927
0.2122 0.2657 Valid: SSE 0.3042 0.2537
0.3444 0.0343 0.2247 0.3539 0.1390
0.2285 0.3491 0.2928 0.1665 0.3181

Epoch 100 : SSE 0.0015 0.0014 0.0016
0.0129 0.0022 0.0021 0.0022 0.0023
0.0018 0.0018 0.0016 0.0021
Epoch 200 : SSE 0.0010 0.0009 0.0010
0.0062 0.0008 0.0006 0.0017 0.0019
0.0009 0.0011 0.0009 0.0012
⋮
Epoch 4900 : SSE 0.0001 0.0000 0.0001
0.0000 0.0000 0.0000 0.0000 0.0001
0.0000 0.0000 0.0000 0.0000
Epoch 5000 : SSE 0.0001 0.0000 0.0001
0.0000 0.0000 0.0000 0.0000 0.0001
```

```
0.0000 0.0000 0.0000 0.0000

Train: SSE 0.0001 0.0000 0.0001 0.0000
0.0000 0.0000 0.0000 0.0001 0.0000
0.0000 0.0000 0.0000 Unseen: SSE
0.0001 0.0001 0.0001 0.0000 0.0000
0.0000 0.0001 0.0001 0.0001 0.0000
0.0000 0.0000 Valid: SSE 0.0000 0.0000
0.0001 0.0000 0.0000 0.0000 0.0001
0.0001 0.0001 0.0000 0.0000 0.0000
```

Fig. 2 Tadpole Plots



4 Discussion

The applications demonstrates that there is some form of underlying relationship between the values presented. The fact that SSE values were able to be learned that low demonstrates this fact. Although only a single value in any instance would have anything more notable than a single value offset for the SSE with respect to the other 12 outputs, the actual computed SSE values are significantly lower than would be expected from this offset. This demonstrates that the network is able to build a successful prediction model.

Increasing the momentum as predicted results in the network not properly training, as each piece of data is no longer evaluated equally.

Increasing the learning rate resulted in the network not training, as it is not finite enough to correctly narrow in on a local minima of the dataset when making predictions.

Changing the number of hidden neurons also resulted in a less efficient network. Fewer resulted in the network not correctly training, and it would reach the point of rising SSE much faster. More resulted in the network taking longer to train and reaching a low SSE slower. This demonstrates the usual rule of $\frac{IO}{2}$ to hold reasonable well.

5 Conclusion

The application results seem to demonstrate that there is some form of clear underlying relationship between predictions of these different linguistic properties.

This would mean that fundamentally, different languages with some features are predictably likely to have other features based on the ones they contain.

This does *not* however demonstrate that languages have some underlying true principles that cause these features to occur together however. As human languages clump by language family, and the validation and unseen data were pulled from natural languages, it could simply be that the neural network was able to correctly predict and classify the family of a language from it's properties.

The network may also be over-training and oversupplied with data. It could be that if it was presented further data pulled from a different source looking to make specific predictions, rather than averaging the error of predictions over bulk data, the network would fail entirely. Instead of trying to find 5 pieces of data from 4 known pieces and 1 unknown, it would be interesting to see if the accuracy can be replicated when trying to find 1 (unknown) piece of data form the 4 known, making the accuracy more significant to ascertain and less reliant on data duplication mapping. It would be interesting to see if the experiment results are as good if it were done as 5 different $4 \rightarrow 1$ (sets) networks instead of a single $5 \rightarrow 5$ (sets) network.

There are also known problems with the accuracy data source WALS[8], however it was hoped that with a large enough volume of data being used, any inaccuracies would be overpowered in the balance of training. In addition to this, the

data may not be entirely accurate to the real world status of a language, it will still represent a form of classification that may occur, and if the data is consistently incorrect this would not effect training.

It would be interesting to be able to run specific input tests on the network post-training. That is a feature that was not available in the software being used, that would provide a more exploratory method of testing and applying the trained neural network.

As a final conclusion, the program was able to build up a prediction model, enabling it to predict relatively accurately state 5 linguistic features from 4 known linguistic features.

REFERENCES

[1] I. Maddieson, "Consonant-vowel ratio," in *The World Atlas of Language Structures Online*, M. S. Dryer and M. Haspelmath, Eds. Leipzig: Max Planck Institute for Evolutionary Anthropology, 2013. [Online]. Available: http://wals.info/chapter/3

[2] M. S. Dryer, "Position of case affixes," in *The World Atlas of Language Structures Online*, M. S. Dryer and M. Haspelmath, Eds. Leipzig: Max Planck Institute for Evolutionary Anthropology, 2013. [Online]. Available: http://wals.info/chapter/51

[3] ——, "Order of object and verb," in *The World Atlas of Language Structures Online*, M. S. Dryer and M. Haspelmath, Eds. Leipzig: Max Planck Institute for Evolutionary Anthropology, 2013. [Online]. Available: http://wals.info/chapter/83

[4] ——, "Order of genitive and noun," in *The World Atlas of Language Structures Online*, M. S. Dryer and M. Haspelmath, Eds. Leipzig: Max Planck Institute for Evolutionary Anthropology, 2013. [Online]. Available: http://wals.info/chapter/86

[5] ——, "Order of adjective and noun," in *The World Atlas of Language Structures Online*, M. S. Dryer and M. Haspelmath, Eds. Leipzig: Max Planck Institute for Evolutionary Anthropology, 2013. [Online]. Available: http://wals.info/chapter/87

[6] A. C. Gimson, *An Introduction to the Pronunciation of English*. New York: St. Martin's Press, 1970.

[7] L. L. Blumire, "cs2nn17-data-processing," Online, 2018. [Online]. Available: https://github.com/LLBlumire/cs2nn17-data-processing

[8] F. Plank, "WALS values evaluated," *Linguistic Typology*, vol. 13, no. 1, jan 2009. [Online]. Available: https://doi.org/10.1515/lity.2009.003