Using Generated LIBS Data as a Base for Neural Network Architecture Development

J. Lehtonen, T. Aaltonen
Satakunta University of Applied Sciences, Pori, Finland
juuso.2.lehtonen@samk.fi, toni.aaltonen@samk.fi

***Abstract* - Laser-Induced Breakdown Spectroscopy (LIBS) is a rapid atomic spectroscopy technique used to measure the concentration of elements in samples. This research paper uses supervised deep learning with a LIBS database provided by National Institute of Standards and Technology (NIST) to find out what kind of neural network architecture can best predict element concentrations from metal alloy samples. Accuracy of predictions was used to evaluate the network architectures. Because collecting and storing real samples of metal alloys with a wide range of element concentrations requires a lot of resources, the network's ability to predict outside of training set's range of concentrations was also tested. The goal of the research was to find out if NIST database together with machine learning can be used to reduce the amount of real alloy samples needed for LIBS calibration.**

***Keywords* - LIBS; NIST; neural network; supervised learning; deep learning; regression**

I. Introduction

In this paper we explore what kind of neural network architecture is best suited to predict element concentrations from laser-induced breakdown spectroscopy (LIBS) data generated from the National Institute of Standards and Technology (NIST) Atomic Spectra Database (ASD) [1]. We also test if the network can predict concentrations outside of the training range. As the generated data is similar to measured data, we try to find the best kind of network without the need for large number of samples and measurements [2].

Several studies have been made with machine learning , LIBS and NIST data [3][4]. Many of the studies done earlier use partial least squares algorithms or support vector machines. NIST data hasn't been widely utilized in training the algorithms and is often used for selecting the spectral lines that are most interesting for the study.

II. Generated LIBS Spectra

The dataset used for the tests was acquired through NIST website using SimulatedLibs [5] with modifications to acquire the spectra faster. The modifications were using a session for the requests and using multiprocessing. Generated LIBS spectrum line intensities are determined by electron temperature, electron density and the elemental composition of the sample. When electron temperature and density stay static the spectral line intensities change linearly (Fig 1.) as a function of elemental concentration when the data is generated on the website [6]. Changing the composition affects the intensities but has little to no effect on the relative

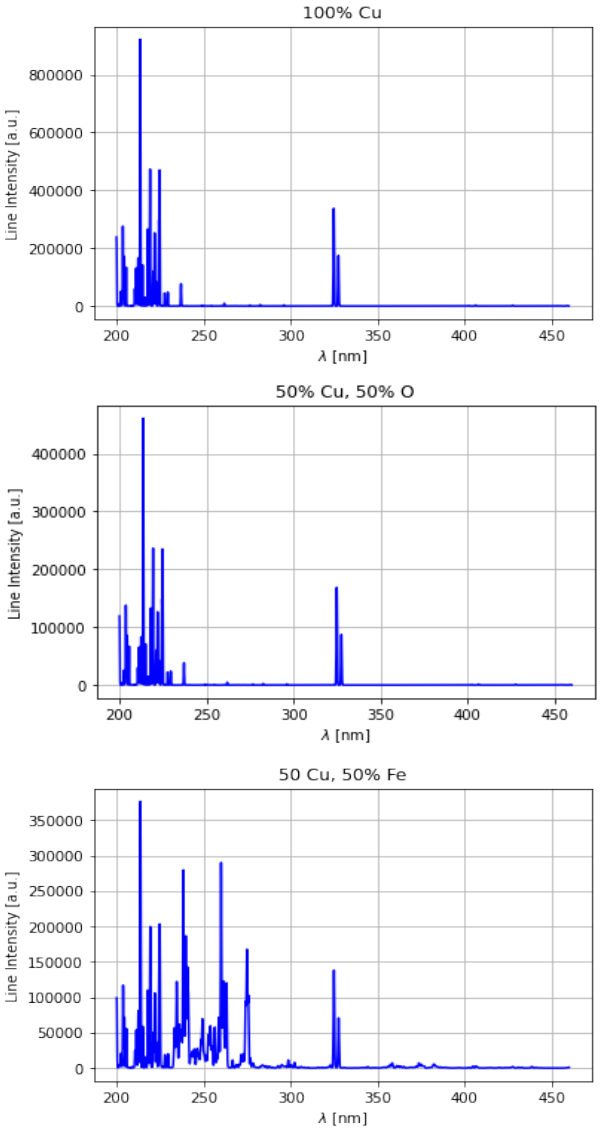

Figure 1: The relative line intensities stay the same when elemental composition changes unless the elements have overlapping lines. The line intensities from 100% copper (Cu) sample to 50% sample are halved and lines from oxygen (O) are not visible in the plot since they are less intense compared to Cu with the chosen electron temperature and density.

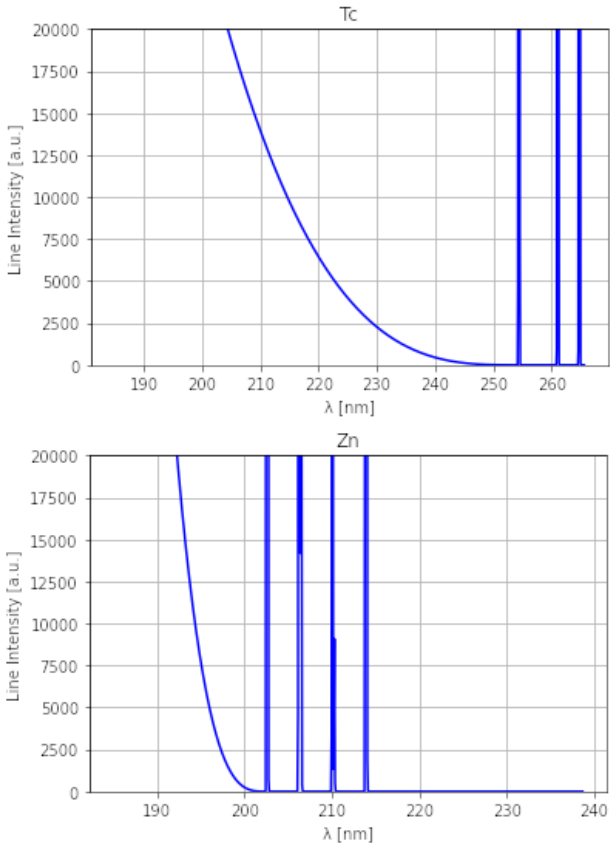

Figure 2: Some of the samples have erroneous data, possibly a result of cubic spline interpolation. The erroneous data is usually found at the start or end of the spectrum.

intensities between lines where elements have no overlapping spectral lines.

The generated dataset consists of 24 elements, C, Mg, Al, Si, Ti, V, Fe, Co, Ni, Cu, Mn, Li, Zn, O, P, Pt, Au, Ag, Ir, W, Pd, Sn, Tc, and Y, with combinations of concentrations from 100% to 1% with two elements, and combinations of three element compositions of [33, 34, 33], [40, 20, 40], [20, 40, 40], [40, 40, 20], [60, 20, 20], [20, 60, 20], [20, 20, 60], [80, 10, 10], [10, 80, 10], [10, 10, 80], [15, 30, 55], [30, 55, 15], [55, 15, 30]. The combinations add up to a total of 53,660 samples. The samples are of electron temperature 1 eV and electron density of 1e17 cm-3. The data was generated between 185 nm and 460 nm wavelengths, with a resolution of 0.067 nm, because a spectrometer used in our other experiments has an average resolution of 0.067 nm. The labels are generated are 24 numbers, one for each element, between 0 and 1.

The data has some abnormalities possibly due to cubic spline interpolation used in the generation process. Some of the samples have “tails” that are not spectral lines (Fig 2.). The data was not cleaned of these abnormalities prior to the tests, and they might affect the results.

III. Methodology

TensorFlow [7] was used to create and train the networks. The networks consisted of two fully-connected hidden layers of varying widths, input layer of 4094 and output layer of 24, one for each of the elements. The hidden layers use ReLU [8] activation function, and the output layer uses linear activation. The linear activation was chosen instead of softmax, because in real measurements there can be elements present which are not included in the datasets. A combination of 16, 32, 64, 128, 256, 512 widths for both hidden layers were tested, the 16 width layers were dropped from the tests after some time because the other sized networks consistently showed significantly better results. Different numbers of epochs and batch sizes were also tested. Epoch sizes from 5 to 40 with a step size of 5 were tested. Tested batch sizes were 2, 4, 8, 16, 32, 64, 128 and 256 [9]. Tests were also run with kernel and/or bias regularization applied to both hidden layers using default settings of TensorFlow. Altogether, more than 12800 networks were trained and tested. The initialization of weights used was the default in TensorFlow, glorot uniform [10]. Adam [11] with the default hyperparameters from TensorFlow was used as an optimizer. The data was randomly divided into 60:40 training:test sets. Mean squared error (MSE) was used to evaluate the networks. The networks were only trained once with each configuration of epochs, batch size and different width layers because of the time it takes to train each network and the number of combinations was already sizable, this results in randomness in measured MSE because of the randomness of weight initialization. Dataset was scaled to 0-1 range, which preserves the relative intensities of the peaks.

For testing how well a network can learn outside of the training range, the test set was changed to copper samples which had less than 70% of copper or more than 90% of copper and the rest of the data was used as a training set. This resulted in a training set with 48829 samples and test set of 4831 samples. This was done to see if a network could be trained to extrapolate outside of the range of real samples, which are difficult to get and store in large numbers.

Inference time of the networks were tested with 1000 samples picked randomly from 2000 samples. A prediction was made for each of the 1000 samples and the time it took to run inference was recorded. These tests were only done for the networks without bias and kernel regularization to reduce the time to run the tests since doing 1000 inferences for a network took approximately 45 to 60 seconds. The tests were run on a GPU-server using a NVIDIA RTX A5000, 10 cores of AMD Epyc 7413 processor and 50 GB random access memory.

IV. Results

The average inference time of 1000 independent predictions for the networks was 52 seconds with a standard deviation of 2.9 seconds, the fastest time being 41.7 seconds and the slowest 82.8 seconds as seen on table 1. The tests might have been affected by some

Table 1: The data sorted by inference time, with 5 of the fastest and slowest results showing. The three slowest times are clearly slower than the rest of the tests.

Epochs	Batch size	First layer	Second layer	Inference time (1000 samples)
5	2	16	16	41.681572 s
5	2	16	32	42.710497 s
5	2	16	64	44.343524 s
5	2	32	16	44.729965 s
5	2	16	256	44.830615 s
…	...	...	...	...
30	4	32	16	64.053025 s
30	4	64	128	64.097452 s
35	4	128	256	67.813859 s
20	128	32	32	72.505119 s
30	4	64	256	82.783849 s

unknown software or hardware issues. The inference time was tested to see if the size of the network matters for practical use. The inference tests were only run with networks not using regularization to save time. The larger networks achieve lower MSE than the smaller networks and benefit more from longer training as seen in table 2. The networks' ability to generalize to the test set also seems to benefit from having a larger batch size as the 100 lowest test MSE only have one network with less than 16 batch size.

Almost all the networks with kernel regularization perform worse than without it, but for most of the networks the test set MSE is lower than minimum and final loss from training. The minimum and final loss are often the same which could mean that the networks with kernel regularization could benefit from more training. The worst MSE achieved was 0.024501 as shown in table 3.

Testing the networks' ability to generalize outside of the training concentration showed that the network can predict outside of training range up to a point. The test was conducted with the best performing network from the first test. The training loss after 40 epochs was 0.00077266 and the test set MSE was 0.004. A considerably larger training set most likely helped to bring the training loss down. The closer the tested values are to the training range of 70-90% the better the predictions are. The largest error in predicting the copper concentration was 85 percentage points, which came from a sample with 10% copper, 80% carbon and 10% phosphorous. In the worst 20 predictions, 13 of the samples have copper and oxygen in them and the other 7 have copper with 2 other elements in them as shown in table 4. All the samples with only copper and oxygen in the bottom 20 predictions have 15% or less copper. 133 predictions out of 4831 are less than 1 percentage point off. When looking at the most accurate predictions for copper we can see that prediction of other elements in those samples was off tens of percentage points in some cases, as shown in table 5. Five of the best performing networks were trained again to see if the performance was consistent. Each of the five

Table 2: 20 networks with the lowest MSE. Larger networks and longer training achieve lower MSE with the test set.

Epochs	Batch size	First layer	Second layer	Kernel regularization	Bias regularization	Minimum loss during training	Final loss	Test set MSE
40	32	512	128	No	No	8.13e-4	8.18e-4	8.77e-4
40	256	512	256	No	No	8.12e-4	8.20e-4	9.15e-4
40	256	512	512	No	Yes	8.02e-4	8.02e-4	9.25e-4
35	64	512	128	No	No	8.33e-4	8.33e-4	9.31e-4
35	128	512	128	No	No	8.43e-4	8.43e-4	9.51e-4
35	128	512	256	No	No	8.03e-4	8.25e-4	9.58e-4
35	32	512	128	No	No	8.35e-4	8.35e-4	9.60e-4
40	64	128	512	No	Yes	8.29e-4	8.29e-4	9.63e-4
25	128	512	512	No	No	8.80e-4	8.80e-4	9.76e-4
40	16	128	512	No	No	8.93e-4	8.93e-4	9.78e-4
40	64	64	512	No	No	8.79e-4	8.83e-4	9.79e-4
35	64	128	512	No	Yes	8.68e-4	8.68e-4	9.80e-4
40	256	512	128	No	Yes	8.89e-4	8.89e-4	9.82e-4
25	128	512	512	No	Yes	8.77e-4	8.77e-4	9.84e-4
30	64	512	128	No	No	8.84e-4	8.96e-4	9.87e-4
30	32	512	512	No	No	8.51e-4	8.56e-4	9.88e-4
30	128	512	256	No	No	8.51e-4	8.51e-4	9.91e-4
40	32	512	256	No	No	7.94e-4	7.94e-4	9.92e-4
40	64	512	256	No	No	7.90e-4	7.90e-4	9.96e-4
25	32	512	512	No	No	9.10e-4	9.10e-4	9.97e-4

Table 3: Networks trained with batch size 2 or 4 perform consistently worse than networks trained with larger batch size.

Epochs	Batch size	First layer	Second layer	Kernel regularization	Bias regularization	Minimum loss during training	Final loss	Test set MSE
15	2	32	64	Yes	No	0.021380	0.021814	0.022002
5	2	512	256	Yes	Yes	0.021901	0.021957	0.022210
30	2	64	512	Yes	Yes	0.021889	0.027678	0.022454
25	2	128	512	Yes	Yes	0.021906	0.027999	0.022508
20	2	128	512	Yes	Yes	0.021904	0.025114	0.023366
25	2	128	256	Yes	Yes	0.021844	0.023886	0.024065
5	2	256	512	Yes	Yes	0.021934	0.021978	0.024501

networks were trained 20 times and the test set MSE was recorded for each run. In figure 3 we can see that the previously third best network is performing better than in the single training run while the order of the other networks has stayed the same. The overall MSE seems to be lower in the new tests, but the code was not changed.

V. CONCLUSION AND FURTHER RESEARCH

Predicting concentration from generated data with a neural network is possible, but gets harder for data that is further away from the provided training range. With the data tested the network size can be quite small, and the inference times are not affected much by the network size at such a small scale. Further research can be done with larger datasets and with varying electron temperature and density. In the future the data should also be cleaned from the anomalies caused by interpolation. The networks could benefit from more epochs. The large prediction error with some of the elements suggests that more careful selection of elements for the dataset could be beneficial when using the generated data as a base for the models used with real samples.

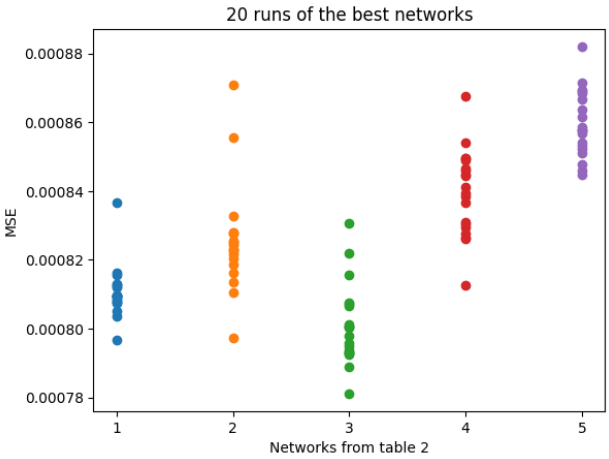

Figure 3: 5 of the best performing networks were trained 20 times each to see how much the results vary. X-axis shows the ranking of the networks from the table 2 tests. Blue dots were the best network in single training run, but the green dots show better results in the tests with multiple training runs.

Table 4: 20 of the predictions with the largest error for copper. Error is calculated by taking an absolute of model prediction of copper – copper label, first value in the element columns are label values and second values are the predictions, both multiplied by 100. Only elements included in the labels are shown.

Error	Cu	O	P	C	Si
0.852328	10 / 95	0 / -8	10 / 4	80 / 49	0 / -2
0.806924	10 / 91	0 / -5	80 / 16	10 / 12	0 / -1
0.799032	15 / 95	0 / -8	30 / 6	55 / 42	0 / -2
0.758974	1 / 77	99 / 21	0 / -3	0 / 0	0 / 1
0.749208	2 / 77	98 / 21	0 / -3	0 / 0	0 / 1
0.745582	20 / 95	0 / -6	40 / 7	40 / 25	0 / -1
0.745178	20 / 95	0 / -8	20 / 5	60 / 42	0 / -2
0.743142	20 / 94	0 / -6	60 / 9	20 / 13	0 /-1
0.739396	3 / 77	97 / 21	0 / -3	0 / 0	0 / 1
0.729482	4 / 77	96 / 21	0 / -3	0 / 0	0 / 1
0.719343	5 / 77	95 / 21	0 / -3	0 / 0	0 / 1
0.709479	6 / 77	94 / 21	0 / -3	0 / 0	0 / 1
0.699579	7 / 77	93 / 21	0 / -3	0 / 0	0 / 1
0.689445	8 / 77	92 / 21	0 / -3	0 / 0	0 / 1
0.679562	9 / 77	91 / 21	0 / -3	0 / 0	0 / 1
0.669623	10 / 77	90 / 21	0 / -3	0 / 0	0 / 1
0.659569	11 / 77	89 / 21	0 / -3	0 / 0	0 / 1
0.649558	12 / 77	88 / 21	0 / -3	0 / 0	0 / 1
0.642394	10 / 74	80 / 23	0 / -7	0 / -2	10 / 45
0.639570	13 / 77	87 / 21	0 / -3	0 / 0	0 / 1

Table 5: 5 of lowest absolute errors for copper prediction with the element label and prediction multiplied by 100. Only elements included in the labels are shown.

Error	Cu	Mg	Mn	Tc	Zn
1.00e-4	63 / 63.00	37 / 26.92	0 / -0.34	0 / -0.98	0 / -0.60
1.47e-4	64 / 63.99	0 / -0.61	36 / 28.13	0 / -2.04	0 / -1.58
2.92e-4	20 / 19.97	20 / 14.25	0 / 0.00	60 / 37.63	0 / -0.69
5.48e-4	61 / 61.05	0 / 0.00	0 / -0.11	0 / -0.99	39 / 29.56
7.54e-4	64 / 63.92	36 / 26.69	0 / -0.30	0 / -0.94	0 / -0.56

REFERENCES

[1] Kramida, A., Ralchenko, Yu., Reader, J. and NIST ASD Team, 2022, NIST Atomic Spectra Database (version 5.10), [Online]. Available: https://physics.nist.gov/asd [Fri Feb 03 2023]. National Institute of Standards and Technology, Gaithersburg, MD. DOI: https://doi.org/10.18434/T4W30F

[2] Poggialini, Francesco & Campanella, Beatrice & Legnaioli, Stefano & Pagnotta, Stefano & Raneri, Simona & Palleschi, Vincenzo. (2020). "Improvement of the performances of a commercial hand-held laser-induced breakdown spectroscopy instrument for steel analysis using multiple artificial neural networks", Review of Scientific Instruments. 91. 073111. 10.1063/5.0012669.

[3] Thomas F. Boucher, Marie V. Ozanne, Marco L. Carmosino, M. Darby Dyar, Sridhar Mahadevan, Elly A. Breves, Kate H. Lepore, Samuel M. Clegg, 2015, "A study of machine learning regression methods for major elemental analysis of rocks using laser-induced breakdown spectroscopy", Spectrochimica Acta Part B: Atomic Spectroscopy, DOI: https://doi.org/10.1016/j.sab.2015.02.003.

[4] Tianlong Zhang, Shan Wu, Juan Dong, Jiao Wei, Kang Wang, Hongsheng Tang, Xiaofeng Yang, Hua Li, 2015, "Quantitative and classification analysis of slag samples by laser induced breakdown spectroscopy (LIBS) coupled with support vector machine (SVM) and partial least square (PLS) methods", DOI: https://doi.org/10.1039/C4JA00421C

[5] MKastek, 2022, SimulatedLIBS v1.0.0 https://github.com/MKastek/SimulatedLIBS

[6] NIST, 03.02.2023, NIST LIBS Database, website source code lines:964,966 , https://physics.nist.gov/PhysRefData/ASD/LIBS/js/saha_lte.js

[7] Abadi et al., 2016, "TensorFlow: A system for large-scale machine learning", DOI: https://doi.org/10.48550/arXiv.1605.08695

[8] Raman Arora, Amitabh Basu, Poorya Mianjy, Anirbit Mukherjee, 2016, "Understanding deep neural networks with rectified linear units", DOI: https://doi.org/10.48550/arXiv.1611.01491

[9] Dominic Masters, Carlo Luschi, 2018, "Revisiting small batch training for deep neural networks", DOI: https://doi.org/10.48550/arXiv.1804.07612

[10] Boris Hanin, David Rolnick, 2018, How to Start Training: The Effect of Initialization and Architecture, DOI: https://doi.org/10.48550/arXiv.1803.01719

[11] Diederik P. Kingma, Jimmy Ba, 2014, "Adam: a method for stochastic optimization", DOI: https://doi.org/10.48550/arXiv.1412.6980