Convolutional Neural Networks Applied to 2D and 3D DC Resistivity Inversion

S. Weit[1], R.-U. Börner[1], M. Brändel[2], P. Gödickmeier[1], R. Gootjes[2], S. Kost[2], O. Rheinbach[2], M. Scheunert[1] and K. Spitzer[1]
[1]Institute of Geophysics and Geoinformatics, TU Bergakademie Freiberg
[2]Institute of Numerical Mathematics and Optimization, TU Bergakademie Freiberg

SUMMARY

A neural network approach has been developed to invert 2D and 3D apparent resistivity data. The network utilizes convolutions as well as pooling and unpooling operations to transform pseudosections into an underground resistivity model. To train the network, synthetic data were produced using an in-house finite element routine. The subsurface models to produce the data consist of homogenous halfspaces with 0 to 5 conductive spherical anomalies per simulated measurement. The anomalies, if present, have a constant total cross-sectional area. The network was trained on 15300 simulated measurements in 2D and 7500 in 3D. Results show a fairly accurate match of anomaly resistivity and location between ground truth and prediction for larger anomalies, while smaller anomalies often blend together in the prediction. The Background resistivity is often overestimated by the network. Due to the way training was performed, the applicability of the network is currently limited to a small number of scenarios. Despite its limitations, the speed of the prediction and the lack of required a-priori information are advantageous. Possible applications of neural networks in DC resistivity inversion lie in the generation of suitable starting models for other, more traditional inversion techniques.

Keywords: Machine Learning, Convolutional Neural Network, DC, Inversion

INTRODUCTION

Geophysical Inversion is often very time consuming and requires previous knowledge on the geological environment of the survey area. Both of these factors could be avoided or alleviated by utilizing neural networks for geophysical inversion. Distinguishing the effects of anomalies from the effects produced by a geologic background is very similar to an image segmentation approach. This approach is very established and has been shown to work in a variety of fields. Image segmentation is traditionally accomplished using convolutional neural networks. Here, we present an approach of utilizing convolutional neural networks for the task of regression applied to DC data. Network architecture, forward modelling and preliminary results will be discussed.

ARCHITECTURE

Architecture for segmentation often follows an encoder/decoder scheme (see Figure 1). During the encoding, convolutional layers and pooling layers are used to filter out the most important information. This information is then distributed and transformed during the decoder step with the use of unpooling layers and more convolutions. The traditional SegNet architecture (Badrinarayanan et al, 2017) uses a varying number of pooling and unpooling layers, with a final activation function (usually softmax) for pixel classification. Here, this architecture is slightly modified to perform regression by replacing the final activation function with a fully connected layer (see Figure 1). This way, not the categories of each cell (anomaly/background) but the value of each cell (resistivity) are predicted. This 2D network also mirrors the approach to an architecture used by Vu and Jardani (2021). To reduce the amount of overtraining and therefore make the network generalize more, dropout layers were introduced into every convolutional block. These set random entries of the input tensor to 0. In the 3D case, the unpooling layer is replaced by a combination of upsampling the output from the previous step as well as concatenating it with the output of the respective layer during the encoder.

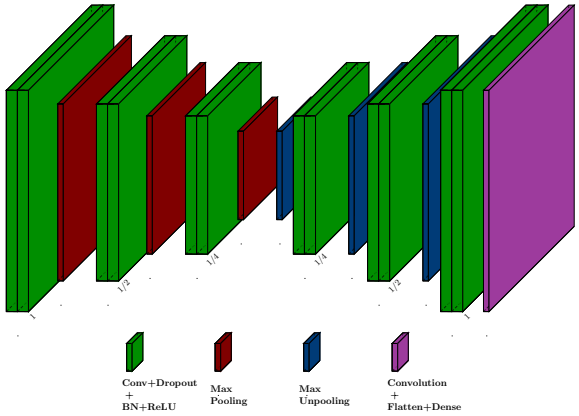

Figure 1: Architecture used for training 2D inversion

This approach effectively turns the network into a U-Net (Ronneberger et al, 2015) for the 3D case. This decision was made due to the lack of an established unpooling approach in 3D that utilizes the positional information of the pooling step. All training and evaluation were performed using the Keras API for Tensorflow (Abadi et al, 2016).

Data Simulation and Preparation

To attain the amount of data needed for training a neural network, synthetic DC data were generated. The synthetic forward modeling was performed using Matlab with an in-house finite element routine (Scheunert et al, 2022). We simulated DC measurements along 17 Profiles with 17 Electrodes each in a dipole-dipole configuration. Each simulated 3D measurement (17 dipole-dipole profiles with 17 electrodes each) will be called a "sample" in the following. The distance between profiles and electrodes was $3\ m$. This setup was deliberately chosen to compare results with Vu and Jardani (2021).

Dataset	n_{train}	n_{eval}
Dataset 1 (base)	7500	1563
Dataset 2 (no anomalies)	800	200
Dataset 3 (varying ρ for BG)	7000	1000

Table 1: Amount of training and evaluation data for each dataset.

For training, 3 sets of data were used: A basic one with spherical anomalies in an otherwise homogenous background of constant resistivity, one dataset with fully homogenous background and one with spherical anomalies in a homogenous background of varying resistivity. The amount of data in each set are shown in Table 1. The 0-5 spherical anomalies were randomly placed, with their centres below the central profile.

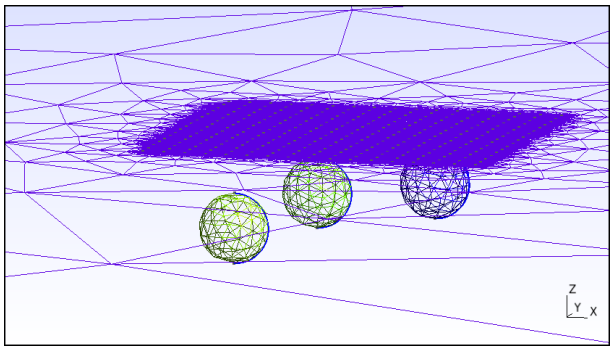

Figure 2: Mesh example for 3 spheres. Mesh is only shown on boundary surfaces.

The sum of the cross sectional area of all spheres was constant among all samples (excluding the homogenous halfspace set). Background resistivity was set to $4,000\ \Omega m$ for the base data set and varied from $4,000 - 40,000\ \Omega m$ for the third data set. Anomaly resistivity varied from $10\ \Omega m$ to $2,000\ \Omega m$. An example with 3 spheres is shown in Figure 2.

The 17 pseudo sections were mapped onto a 32x32x16 grid with grid cells of the size 1.5m in every spatial direction (see Figure 3 for one profile). Background resistivity was assumed for every cell not corresponding to a value from the pseudo section (again note Figure 3).

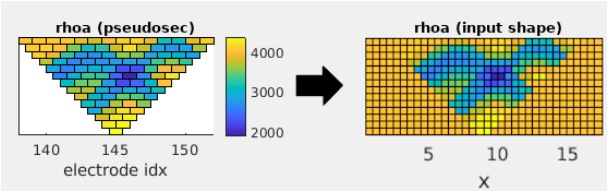

Figure 3: Mapping of a pseudo section to a 32x16 grid.

Inputs ρ_a and ρ were then finally transformed before being supplied to the network:

$$\rho_{a,in} = 10(log_{10}(\rho_a) - 3.5) \quad (1)$$

$$\rho_{in} = log_{10}(\rho) \quad (2)$$

Noisy data were created by the addition of normally distributed random numbers ($\mu = 0$ and $\sigma = 75$) onto every grid point value belonging to the pseudosection.

RESULTS

Figure 4 shows the 2D network prediction for data from all 3 datasets by a network that was trained using samples from all 3 datasets. The samples themselves have not been part of the training but were separated before. Structures close to the surface can be resolved fairly accurately for both cases with anomalies. Deeper structures often blend together or are not resolved at all.

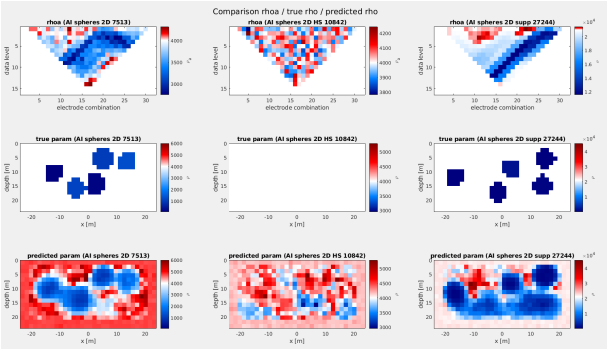

Figure 4: Results for image regression in 2D with all 3 datasets. Top represents the input grid, middle the ground truth, bottom the prediction.

This can be expected due to deeper structures being masked by shallower structures in the data. The background resistivity is generally wrong across evaluations and often shows small-scale variations not present in the ground truth. This is particularly visible in the evaluation of the pseudosection for the homogenous halfspace. There, a wide range of resistivities is predicted, with a great amount of local, small-scale variation, despite none of it being present in the ground truth. This effect can result from a variety of factors, such as small network complexity, dataset structure, noise level, amount of data, etc.

Figure 5 shows network prediction in 3D for the first dataset. A lot of the same effects as in the 2D case can be observed (missing resolvability of deeper anomalies, strong local variation in background resistivity) for this case as well. The general resolvability for smaller structures is worse than in the 2D case.

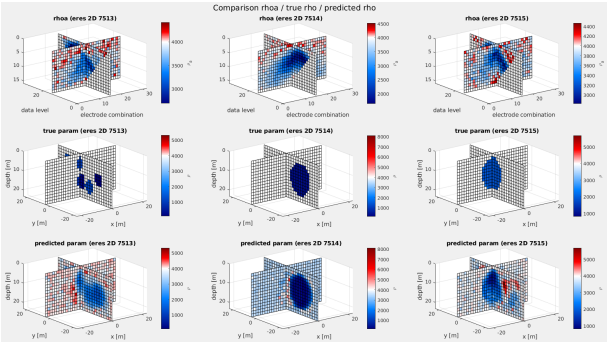

Figure 5: Results for image regression in 3D with the base dataset.

This could be attributed to the fact that a smaller percentage of underground cells belongs to an anomaly compared to the 2D case. Larger Scale anomalies (such as the right and center cases in Figure 5), are resolved well with regards to their position and resistivity.

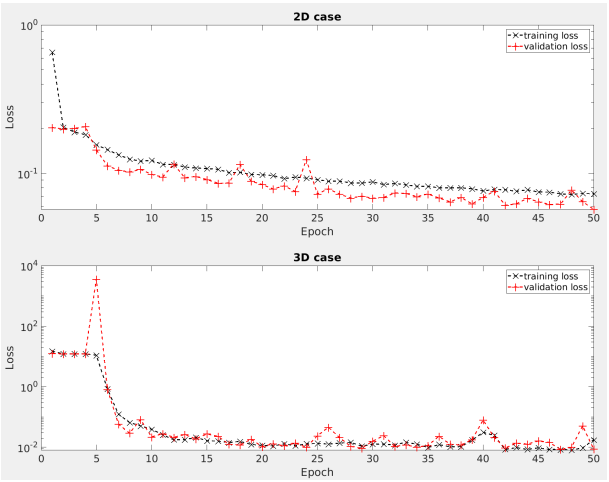

Figure 6: Loss for the training of 2D (above) and 3D networks (below).

Figure 6 shows the loss during training and validation. The curves showcase a small plateau during the start of training, followed by a large drop in loss during a few iterations . After around 10 iterations, improvement of the loss is small for the rest of the training process in 3D. No overtraining (divergence of training and validation loss) can be noticed in either case.

A limitation of the approach is illustrated in Figure 7. There, data from a subsurface model with a more resistive anomaly leads to a prediction that does not match the ground truth.

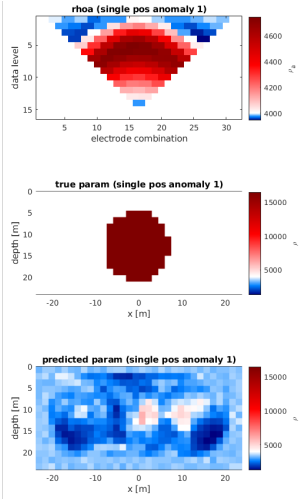

Figure 7: An example of an untrained scenario being evaluated poorly.

As there are no samples covering this case present in the dataset, the network is not able to evaluate cases like these correctly. This illustrates the high dependency on the training data for this type of network.

Conclusions

The presented approach shows promising results for resolving large to medium sized spherical anomalies. Huge limitations lie in the evaluation of scenarios that have not been used for training, such as positive resistivity anomalies. Some of these limitations could potentially be overcome by training with more data and data covering more scenarios. A strong dependency on training data will always be present for this type of network and therefore, alternative approaches should be considered to attain stronger generalization.

References

Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, Devin M, Ghemawat S, Irving G, Isard M, et al (2016) Tensorflow: A system for large-scale machine learning. In: 12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16), pp 265–283

Badrinarayanan V, Kendall A, Cipolla R (2017) Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE transactions on pattern analysis and machine intelligence 39(12):2481–2495

Ronneberger O, Fischer P, Brox T (2015) U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention, Springer, pp 234–241

Scheunert M, Blechta J, Börner RU, Ernst O, Spitzer K (2022) A matlab fe library for the simulation and inversion of em problems. In: 29. Schmucker-Weidelt-Kolloquium für Elektromagnetische Tiefenforschung, Deutsche Geophysikalische Gesellschaft e. V., pp 132–132

Vu M, Jardani A (2021) Convolutional neural networks with segnet architecture applied to three-dimensional tomography of subsurface electrical resistivity: Cnn-3d-ert. Geophysical Journal International 225(2):1319–1331