Deriving groundwater major ions from electrical conductivity using artificial neural networks

Model Development:

To determine the optimal network architecture, the dataset was used to train the model, and eventually the following procedure was established. Two submodels were applied (Figure 2). Submodel 1 included the samples of $EC_{20} > 900$ µS/cm whereas submodel 2 had the samples with lower EC values.

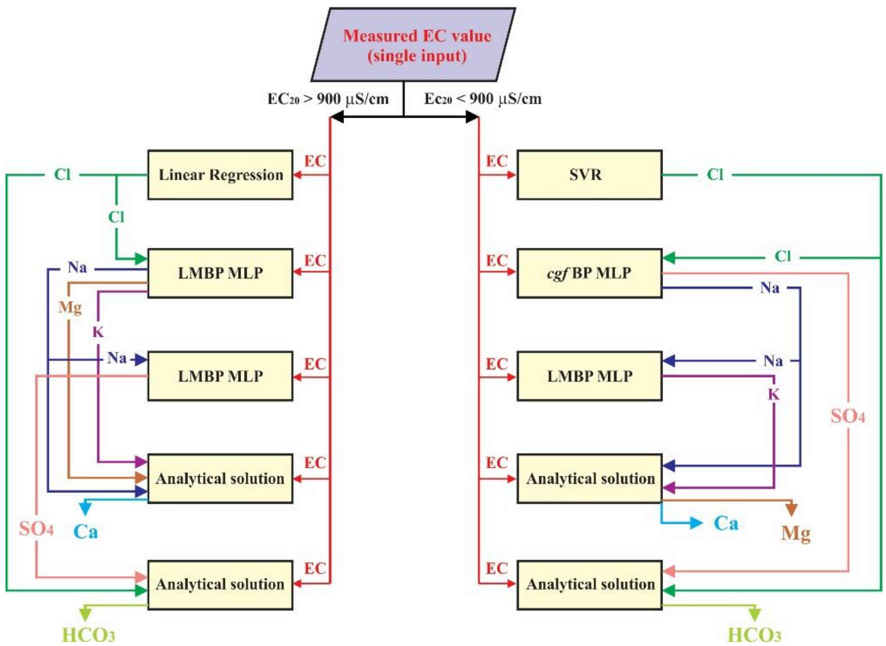

Figure 2. The scheme of the proposed AI model.

Submodel 1 had the following procedure: (1) the Cl was determined from the given measured EC value, (2) Na, K and Mg were then determined from the measured EC and the calculated (predicted) Cl, (3) SO_4 was determined from the EC and the predicted Na, and (4) subsequently Ca and HCO_3 were determined from the following equations (Stuyfzand, 2017) using the aforementioned predicted Na, K, Mg, SO_4 and Cl values besides the measured EC.

$$Ca = (EC_{20} - 33.3645 - 4.0179\ Na - 3.14175\ K - 9.008415\ Mg) / 3.4515 \tag{4}$$

$$HCO_3 = (EC_{20} - 66.7654 - 2.7689\ Cl - 1.4788\ SO_4 - 1.1456\ NO_3) / 1.1142 \tag{5}$$

$$Ca = (EC_{20} /100) - Na - K - Mg \tag{6}$$

$$HCO_3 = (EC_{20}/100) - Cl - SO_4 - NO_3 \tag{7}$$

$$Ca = \sqrt[0.9058]{\frac{EC_{20}}{133.605}} - Na - K - Mg \tag{8}$$

$$HCO_3 = \sqrt[0.9058]{\frac{EC_{20}}{133.605}} - SO_4 - NO_3 \tag{9}$$

Equations 4 and 5 apply for EC_{20} between 300 and 3000 µS/cm (i.e. fresh to slightly brackish waters), and concentrations are in mg/L. Equations 6 and 7 apply for EC_{20} < 300 µS/cm (like most rain waters) whereas equations 8 and 9 apply for EC_{20} > 3000 µS/cm (i.e. saline solutions). Concentrations for equation 6, 7, 8 and 9 are in meq/L, and the NO_3 value is assumed as 7 mg/L or 0.11 meq/L in submodel 1 but 1 mg/L or 0.02 meq/L in submodel 2.

For submodel 2, (1) Cl was determined from the measured EC value, (2) Na and SO_4 were determined from the measured EC and the predicted Cl, (3) K was determined from the measured EC and the predicted Na, and (4) Ca, Mg and HCO_3 were determined based on equations 4 to 9 using the predicted Na, K, SO_4, Cl values, measured EC.

The training of the submodels relied on actual values. The dataset for each submodel was divided into two subsets: training (70%) and testing (30%). The training dataset calibrated the built model. It helped determine the best fitting weights and biases of the network (Haghiabi et al., 2018), and subsequently the optimal structure. On the other hand, the test dataset was used to validate and confirm the prediction accuracy to ensure an accurate performance (Puig-Arnavat and Bruno, 2015). After determining the optimal network and in order to predict the parameters of the whole dataset, the network was run on the predicted values from the previous models and the actual EC.

MATLAB R2020a is used to develop and train all neural network models. Besides, Python is used for building regression and SVR models,

Some Models Architectures:

Submodel 1:

Figure 1 shows the final neural network architecture for predicting Na, Mg, and K for salinized samples. It included 1 hidden layer of 2 neurons with sigmoid activation function and Levenberg-Marquardt back propagation algorithm. The output layer had purelin transfer function.

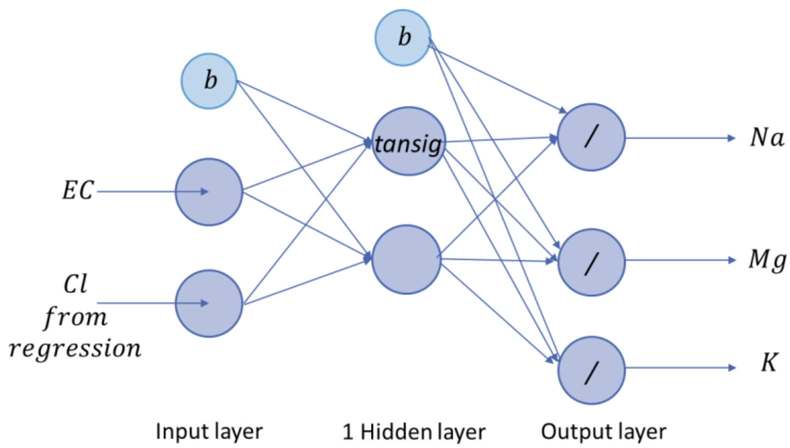

Figure 1

To predict SO_4 , the network (Figure 2) consisted of two hidden layers of 5 and 10 neurons respectively with Levenberg-Marquardt back propagation algorithm.

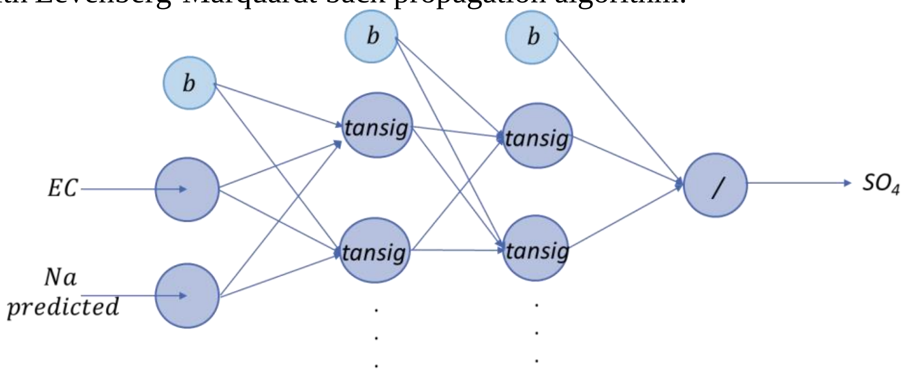

Figure 2

Submodel 2:

Figure 3 shows the neural network for predicting Na, and SO_4 for freshwater samples. It included 1 hidden layer of 20 neurons with sigmoid activation function and *traincgf* back propagation algorithm.

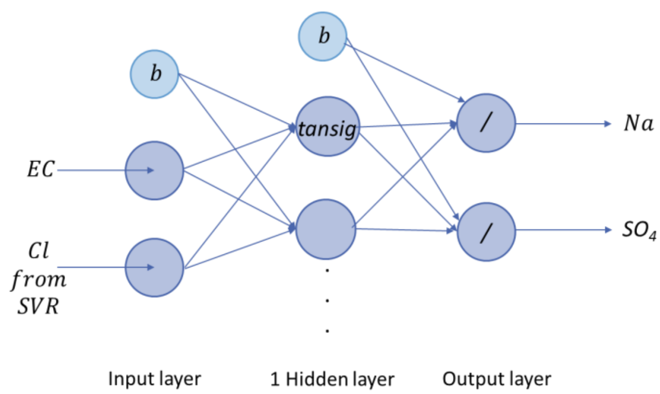

Figure 3

Figure 4 shows that predicting K was possible via a neural network made up of 1 hidden layer of 3 neurons.

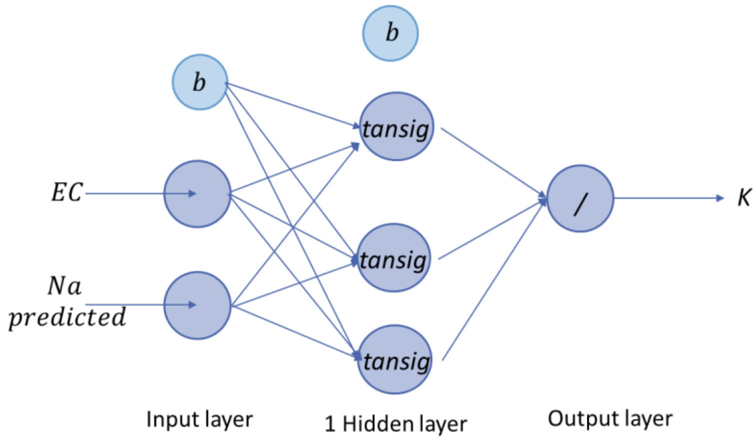

Figure 4

Code Snippets:

1. Building Neural Networks:

```
nb_layers=1;
nb_neurons=2;
algorithm='trainlm'
net = feedforwardnet(2, algorithm);
net.layers{1}.transferFcn = 'logsig';
net.performFcn = 'mse';
net.trainParam.showWindow = true;
```

```
train_fraction = 0.7;
test_fraction = 1 - train_fraction;

% Shuffle your data randomly
data = df(randperm(size(df, 1)), :);

[trainData,testData] = dividerand(data',train_fraction,test_fraction);

trainData = trainData';
testData = testData';

% Separate your inputs and outputs from the training and testing data
X_train = trainData(:, 2:3); % all columns except the last one
X_test = testData(:, 2:3); % the last column
Y_train = trainData(:, 4:end);
Y_test = testData(:, 4:end);

[net, tr] = train(net, data(:,2:3)', data(:,4:6)');
```

2. Analytical Solutions:

```python
def Ca_calculator(EC, Na, Mg, K):
    if EC>=300 and EC<=3000:
        Ca=(EC-33.3645-(4.0179*Na)-(3.14175*K)-(9.008415*Mg))/3.4515
        return Ca

    Na_meq=Na/22.99
    Mg_meq=(2*Mg)/24.312
    K_meq=K/39.1

    if EC <300:
        Ca_meq = (EC/100)-Na_meq-K_meq-Mg_meq
        Ca_mg = 20.04*((EC/100)-Na_meq-K_meq-Mg_meq)
        return Ca_mg
    if EC >3000:
        r=EC/133.605
        rr=r**(1/0.9058)
        Ca_meq=rr-Na_meq-K_meq-Mg_meq
        Ca_mg=20.04*(rr-Na_meq-K_meq-Mg_meq)
        return Ca_mg
```

```python
def HCO3_calculator(EC, Cl, SO4):
    NO3=7.2
    if EC>=300 and EC<=3000:
        hco3= (EC-66.7654-(2.7689*Cl)-(1.4788*SO4)-(1.1456*NO3))/1.1142
        return hco3

    Cl_meq=Cl/35.453
    SO4_meq=(2*SO4)/96.06
    NO3_meq=NO3/62

    if EC<300:
        hco3_meq= (EC/100) - Cl_meq - SO4_meq - NO3_meq
        hco3_mg= 61.02*((EC/100)- Cl_meq - SO4_meq - NO3_meq)
        return hco3_mg

    if EC>3000:
        r=EC/133.605
        rr=r**(1/0.9058)
        hco3_meq=rr-SO4_meq - NO3_meq - Cl_meq
        hco3_mg=61.02*(rr-Cl_meq-SO4_meq-NO3_meq)
        return hco3_mg
```

References

Haghiabi, A. H., Nasrolahi, A. H., & Parsaie, A. (2018). Water quality prediction using machine learning methods. *Water Quality Research Journal*.

Puig-Arnavat, M., & Bruno, J. C. (2015). Chapter 5 - Artificial Neural Networks for Thermochemical Conversion of Biomass. In *Recent Advances in Thermo-Chemical Conversion of Biomass* (pp. 133-156). Elsevier. doi:https://doi.org/10.1016/B978-0-444-63289-0.00005-3

Stuyfzand, P.J., 2017. Hydrogeochemcal (HGC 2.7), for storage, management, control, correction and interpretation of water quality data in Excel® spread sheet. KWR report BTO 2012. 244(s), Nieuwegein.