Tackling earthquake detection problem using 1D Convolutional Neural Networks

Josipa Majstorović, Sophie Giffard-Roisin and Piero Poli

5èmes Rencontres Scientifiques et Techniques Résif, 17 November 2021, Obernai

josipa.majstorovic@univ-grenoble-alpes.fr
Basics of neural networks, deep learning models

Developing CNN detector

What does interpretation stand for in DL?

**Goal:** to further understand how we can exploit the existing DL models for the earthquake detection and whether we can find some other application.
**NEURAL NETWORK**

**BASIC UNIT - NEURON (NODE)**

During the training process of the neural network we adjust the weights and biases.

---

Deep learning - Collection of neural networks, biologically inspired networks, that extract abstract features from the data in a hierarchical fashion.
The performance of the NN model highly depends on:

1. **Training data**
   - Datasets and labels!

2. **Training process**
   - Hyperparameters such as optimising algorithms, learning rates …

3. **Modelling approach**
   - Classification vs regression

4. **NN architecture**
   - Hyperparameters such as nb. of neurons, nb. of layers…

*M. Mousavi presentation*

**Decision making process**
Introduction

Basics of neural networks, deep learning models

Developing CNN detector

*Majstorović et. al., 2021, JGR*
Creating a database using one station approach.

After a grid search analysis, the best combination of the training hyperparameters are selected.

Convolutional neural network (CNN)

65k positive samples + 65k negative samples

Majstorović et al., 2021, JGR
Convolutional neural network (deep learning) - supervised training process

**Convolutional layers**

**Pooling layers**

**Fully-connected layers**

**Outline**
1. Training data
2. Training process
3. Modeling approach
4. NN architecture

**Majstorović et al., 2021, JGR**

**Input Layer**
(number of grid points)

**Hidden Layers**
(2 layers, 8 nodes each)

**Output Layer**
(2 nodes)

**Convolutional layer**

**Pooling layer**

**Fully-connected layer**

**Legend**
- Blue: Convolutional layer
- Green: Fully connected layer
- Stride + Padding + ReLu
- ReLu
- Sigmoid/Softmax

**Detector prediction:**
one-unit vector

**36k of trainable parameters**
Introduction

1. Training data
2. Training process
3. Modelling approach

Outline

Introduction

CNN detector

Interpretation

Convolutional neural network (deep learning) - supervised training process

Convolutional layers

3 channels x 500 samples

Input array

Filter (kernel) size

3 x 3 x 32

Feature map (251 features)

32 channels

Filter (kernel) size

3 x 32 x 32

Feature map (126 features)

32 channels

Majstorović et. al., 2021, JGR
Introduction

Basics of neural networks, deep learning models

Outline

1. Training data
2. Training process
3. Modelling approach
4. NN architecture

CNN detector

Developing CNN detector

Interpretation

What does interpretation stand for in DL?

Majstorović et. al., 2021, JGR
ML (NN, DL) models are often criticised by end users as being a “black box” because of the perceived inability to understand how ML makes its prediction.

Linardatos, 2020, Entropy
The weights and biases that define the kernels (the filters) and that are being adjusted in the training process are now frozen to investigate how those impact the input data.

The training of the DL model is a stochastic process - repeating the training process with the same dataset and the same hyperparamters yields different weights and biases.
Kernel visualisation

Feature map visualisation
Introduction

CNN detector

Interpretation

Outline

Kernel visualisation

Feature map visualisation

Backward optimisation

Layer-wise relevance method

Feature visualisation
Kernel visualisation

Feature map visualisation
**Outline**

- Introduction
- CNN detector
- Interpretation

**Feature visualisation**

- Backward optimisation
- Layer-wise relevance method

---

**Z component**

<table>
<thead>
<tr>
<th>Increasing distance</th>
<th>Y axis is the same per column/channel</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>0.25</td>
<td>0.25</td>
</tr>
<tr>
<td>0.2</td>
<td>0.2</td>
</tr>
<tr>
<td>0.15</td>
<td>0.15</td>
</tr>
<tr>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>-0.05</td>
<td>-0.05</td>
</tr>
<tr>
<td>-0.1</td>
<td>-0.1</td>
</tr>
<tr>
<td>-0.15</td>
<td>-0.15</td>
</tr>
<tr>
<td>-0.2</td>
<td>-0.2</td>
</tr>
<tr>
<td>-0.25</td>
<td>-0.25</td>
</tr>
<tr>
<td>-0.3</td>
<td>-0.3</td>
</tr>
<tr>
<td>-0.35</td>
<td>-0.35</td>
</tr>
<tr>
<td>-0.4</td>
<td>-0.4</td>
</tr>
<tr>
<td>-0.45</td>
<td>-0.45</td>
</tr>
<tr>
<td>-0.5</td>
<td>-0.5</td>
</tr>
</tbody>
</table>

---

**15**
2. We either use
   a) zero,
   b) random,
   c) real input
   and forward propagate it to the output layer.

1. We freeze weights and biases of hidden layers.

3. We calculate $f_{\text{LOSS}}$ by setting desired output to 1 for investigating the earthquake optimal input or 0 for the noise one.

4. The value $f_{\text{LOSS}}$ is backpropagated to update the input.

5. The forward and backward iterations are continued until $f_{\text{LOSS}}$ converges.
1. We freeze weights and biases of hidden layers.

2. We input real earthquake sample.

3. Using the frozen weight and biases we obtain the network output by the forward propagation.

4. We calculate the relevance \( R \) from the previous layer using the relevance propagation rule called the LRP-\( \beta \).

5. We propagate the relevance from the hidden layers up to the input layer.

\[
R_{i=j}^{l,l+1} = \left( 1 + \beta \right) \frac{z_{ij}^+}{z_{ij}^+} - \beta \frac{z_{ij}^-}{z_{ij}^-} R_{i+1}^j
\]

\( \beta = 0 \) - we propagate only positive relevance
\( \beta = 1 \) - we propagate only negative relevance
The propagation rule LRP-β-0
Take away message: there are endless options to explore data with the interpretation methods.
Take away message: there are endless options to explore data with the interpretation methods.

Thank you for your attention!