Brain disease classification using multi-channel 3D convolutional neural networks

Mid-Term Thesis Seminar

Andreas Christopoulos Charitos
Supervisor : Anders Eklund

LINKÖPING UNIVERSITY

LINKÖPING UNIVERSITY

LINKÖPING UNIVERSITY

Welcome back!

What will be presented?

1. Overview of research questions
2. The methods that will be used in the simulations
3. Results so far
4. Critic on the methods and results so far

But first let's take a step back and discuss the data again

Data Overview

- 1035 fMRI images of subjects with ASD and CON
- each image processed to remove time component with different methods resulting different images (derivatives)
- resulting images are 3D volumes of size $(61 \times 73 \times 61)$ except dual regression derivative which has $(61 \times 73 \times 61 \times 10)$
- in total for every subject we have 19 3D image volumes those will be the inputs to our neural network

Lets see a video of a brain volume from ReHo derivative for a subject with ASD and image taken from Yale image center

Questions of interest

During the project we interested in answering the following questions:

Role of derivatives

What is the relation of the derivatives in the classification? We are interested to know if a certain derivative or derivative combination gives better classification

Data augmentation

What kind of 3D augmentation (e.g. rotations, scaling of each volume) are most important in the 3D CNN training?

Architecture

What is the best architecture of the 3D CNN? We wish to know how many layers to use and number of filters in each layer in the model architecture.

Questions of interest

- In the next section we present the methods that will used to answer the questions of interest
- More specifically how we can combine different derivatives to form multi-channel input data
- Which augmentation methods are appropriate for the case study
- How does the search of the best architecture is performed

LINKÖPING UNIVERSITY

Multi-channel Data Construction

- Ensemble learning is a powerful method, inspired by the wisdom of the crowd concept
- Inspired from this concept we wish to combine the available derivatives and construct new datasets
- How? Concatenate the 3D volumes using a 4th channel axis and create multi-channel volumes
- For example, to combine alff ($H(61) \times W(73) \times D(61)$) + dual regression ($H(61) \times W(73) \times D(62) \times C(10)$) we expand the axis of alff valumes adding one extra dimension and concatenate the volumes of each derivative on the channel (4th) axis
- Many combinations which to choose? No rule of thumb so try some at random but one derivative might be more useful (ReHo)
- After the construction the data are split to train/validation/test with ratios 70/15/15 and normalized to [0,1]

Data augmentation

- The datasets that previously constructed still have the same size as the original dataset (1035 images) we need more data for deep CNNs
- Data augmentation is a method to artificially create more data
- CNNs are translation invariant to translations so geometric transformations can be used. Also deformations are easy to implement
- In this study we use the following augmentation methods
  - rotation
  - flip
  - center crop
  - blur
  - elastic deformation

Grid search

- In order to find a proper architecture we need to test some hyper-parameters
- Use grid search to accomplish this task, swap parameters at random
- But range of values might not be finite or bounded and that leads to infinite search space, ouch!
- So define a search space (matrix) with discrete values to test

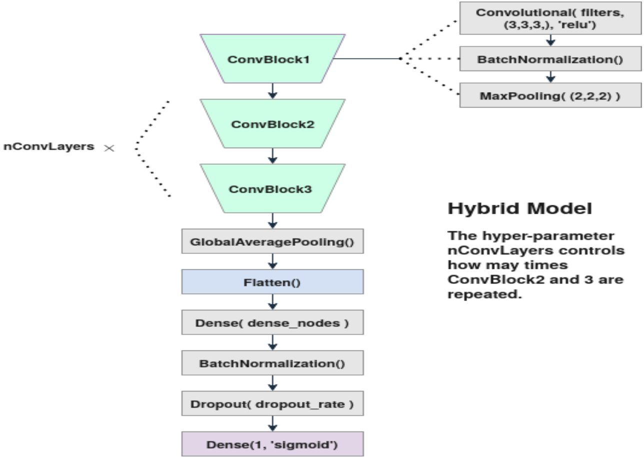
Grid search

- The hyper-parameters of interest are :
  - number of convolutional layers - nConv
  - number of filters in the convolutional layers - nFilt
  - dropout rate - dropRate
  - dense node - denseNum
- After the definition of the search space we define the models overall structure using a hybrid structure HNet
- This structure consists of convolutional blocks. A convblock is simply a convolutional layer followed by a batch normalization and maxpooling layer

Grid search

- One introduction convblock and then 2 convblocks that repeated a number of times according to nConv parameter
- then a globalAveragePooling and a flatten layer
- next densely connected layers, dense layer with nodes updated according to denseNum with batch normalization and a dropout layer with rate updated by dropRate parameter
- finally a dense layer with 1 node to produce final binary label (ASD/CON)

Grid search



Hybrid Model

The hyper-parameter nConvLayers controls how may times ConvBlock2 and 3 are repeated.

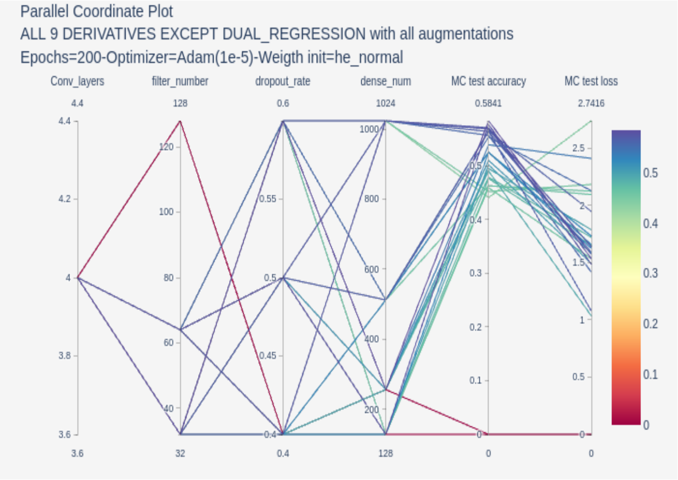
Evaluation

- After the model has trained we evaluate on a hold out dataset
- The traditional evaluation on the hold out dataset is deterministic, we need to test the uncertainty of the model
- Monte Carlo dropout *(Gal and Ghahramani 2015)*, offers a more probabilistic (Bayesian) evaluation method
- In simple words MC dropout is performed by evaluating the model on the test data for a defined number of times. That gives a vector of values for a defined metric (accuracy most common).
- Then calculate mean accuracy of vector as final metric

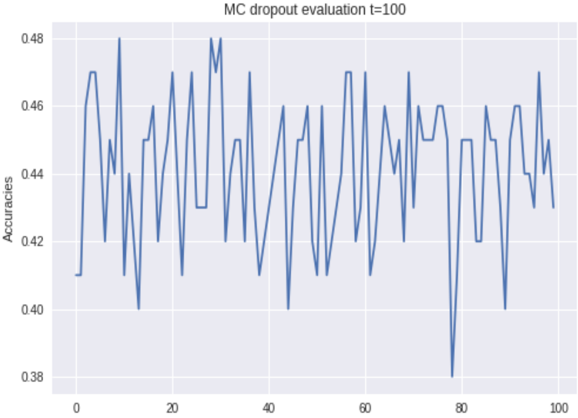
LINKÖPING UNIVERSITY

- As we mentioned before we can create a lot of combinations of 9 derivatives (9!)
- So we choose 3 combinations of interest to test and evaluate
- One combination with only the dual regression derivative (10 channels), one with all the derivatives except dual regression (9 channels) and finally one combination with all the derivatives (19 channels)
- Next, we present some combinations tested with smaller settings

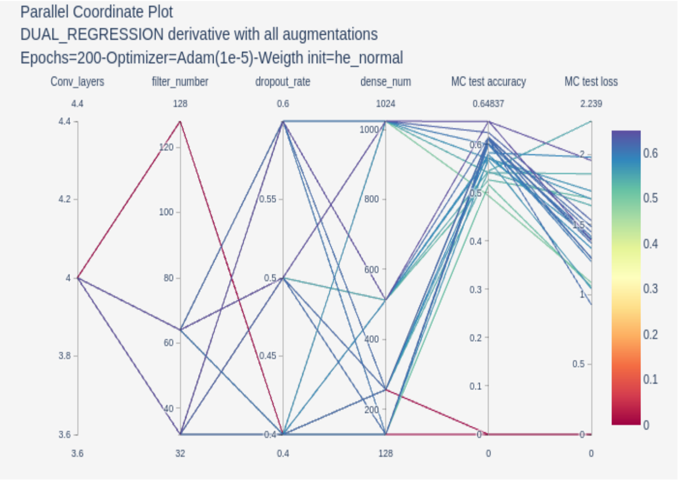
All 9 derivatives except Dual Regression



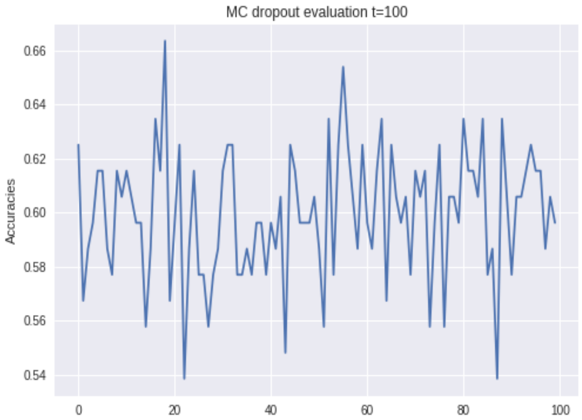
MC evaluation for all 9 derivatives except Dual Regression



Dual Regression Derivative



MC evaluation for Dual Regression derivative



LINKÖPING
UNIVERSITY

What we have achieved so far?

Train and Evaluate

Test some derivative combinations using all augmentations and evaluate the performance using MC dropout

What is next?

Future work

Test the combinations of interest with all and some augmentations. Try to investigate the performance with another activation function, PRELU *(He et al. 2015)* has proven to be very effective with CNNs.

References

Gal, Yarin and Zoubin Ghahramani (June 2015). "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning". In: *Proceedings of The 33rd International Conference on Machine Learning*.

He, Kaiming et al. (2015). "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification". In: *CoRR* abs/1502.01852. arXiv: 1502.01852. URL: http://arxiv.org/abs/1502.01852.

Tack för idag!

www.liu.se

LINKÖPING UNIVERSITY