Simulating the ground truth

Practical Course: Hands-on Deep Learning for Computer Vision and Biomedicine

Tim Kerschbaumer

[1]Department of Informatics
Technichal University of Munich

2017-08-28

Table of Contents

Table of Contents

Motivation

- Supervised learning requires ground truth output targets for training and validation

Motivation

- Supervised learning requires ground truth output targets for training and validation
- In many applications the ground truth cannot be directly obtained

Motivation

- Supervised learning requires ground truth output targets for training and validation
- In many applications the ground truth cannot be directly obtained
- One field where data and labels are expensive to obtain is in diffusion MRI

Motivation

- Supervised learning requires ground truth output targets for training and validation
- In many applications the ground truth cannot be directly obtained
- One field where data and labels are expensive to obtain is in diffusion MRI
- Is it good enough to build and train models from computer simulations of the ground truth?

Diffusion MRI

- Diffusion is the microscopic movement of atoms and molecules in a solution or gas.

Diffusion MRI

- Diffusion is the microscopic movement of atoms and molecules in a solution or gas.
- Molecules such as water floats freely through our bodies

Diffusion MRI

- Diffusion is the microscopic movement of atoms and molecules in a solution or gas.
- Molecules such as water floats freely through our bodies
- In some medical conditions, such as stroke, these molecules can become restricted.

Diffusion MRI

- Diffusion is the microscopic movement of atoms and molecules in a solution or gas.
- Molecules such as water floats freely through our bodies
- In some medical conditions, such as stroke, these molecules can become restricted.
- With diffusion MRI we measure the ability of these molecules to move freely within each voxel.

Diffusion MRI

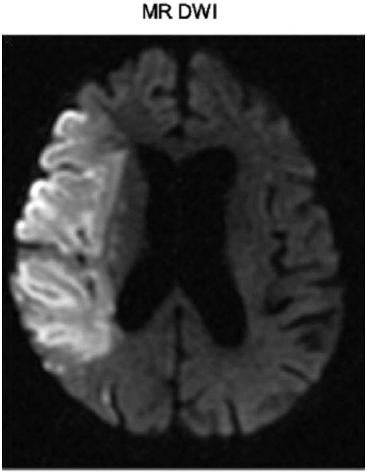


Figure: White area shows restricted diffusion

Table of Contents

Generating data

How to generate data?

Generating data

How to generate data?

- Camino Toolkit - Software tool for Diffusion MRI processing and simulation

Generating data

How to generate data?

- Camino Toolkit - Software tool for Diffusion MRI processing and simulation
- Input: Simulation configuration, i.e cylinder radius, the restricting environment where diffusion takes place.

Generating data

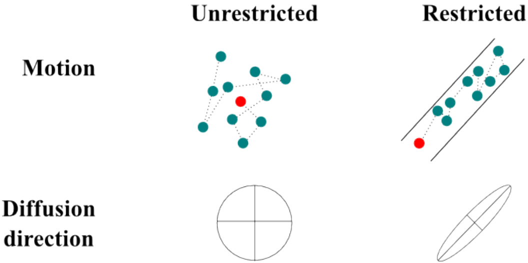
How to generate data?

- Camino Toolkit - Software tool for Diffusion MRI processing and simulation
- Input: Simulation configuration, i.e cylinder radius, the restricting environment where diffusion takes place.
- Output: A list of individual voxels with 288 channels (DWI signal intensity)

Generating data

How to generate data?

- Camino Toolkit - Software tool for Diffusion MRI processing and simulation
- Input: Simulation configuration, i.e cylinder radius, the restricting environment where diffusion takes place.
- Output: A list of individual voxels with 288 channels (DWI signal intensity)



Generating data

What data to generate?

Generating data

What data to generate?

- Want similar to HPC (Human Connectome Project) - Diffusion MRI dataset from scanned persons

Generating data

What data to generate?

- Want similar to HPC (Human Connectome Project) - Diffusion MRI dataset from scanned persons
- Idea: Try wide spread of settings with Camino, use kNN to compare with HPC data.

Generating data

What data to generate?

- Want similar to HPC (Human Connectome Project) - Diffusion MRI dataset from scanned persons
- Idea: Try wide spread of settings with Camino, use kNN to compare with HPC data.
- Run many simulations with these settings.

Data

- Each input is a voxel with 288-channels, i.e a $1 \times 1 \times 1 \times 288$ dimensional array.

Data

- Each input is a voxel with 288-channels, i.e a $1 \times 1 \times 1 \times 288$ dimensional array.
- Can simply be seen as a 288-dimensional feature vector, ignoring spatial $1 \times 1 \times 1$ dimensions.

Data

- Each input is a voxel with 288-channels, i.e a $1 \times 1 \times 1 \times 288$ dimensional array.
- Can simply be seen as a 288-dimensional feature vector, ignoring spatial $1 \times 1 \times 1$ dimensions.
- Output is single value, cylinder radius, the input to each simulation.

Data

- Each input is a voxel with 288-channels, i.e a $1 \times 1 \times 1 \times 288$ dimensional array.
- Can simply be seen as a 288-dimensional feature vector, ignoring spatial $1 \times 1 \times 1$ dimensions.
- Output is single value, cylinder radius, the input to each simulation.
- Simulating data with Camino very slow, around 24 hours for 1000 voxels (with same setting) $\implies$ only have 93900 voxels.

Data

- Each input is a voxel with 288-channels, i.e a $1 \times 1 \times 1 \times 288$ dimensional array.
- Can simply be seen as a 288-dimensional feature vector, ignoring spatial $1 \times 1 \times 1$ dimensions.
- Output is single value, cylinder radius, the input to each simulation.
- Simulating data with Camino very slow, around 24 hours for 1000 voxels (with same setting) $\implies$ only have 93900 voxels.
- Simulated data divided into training 60%, validation 20% and test 20%.

Developing the network

Network considerations

Developing the network

Network considerations

- Different feed-forward architectures

Developing the network

Network considerations

- Different feed-forward architectures
- L1 or L2 loss function

Developing the network

Network considerations

- Different feed-forward architectures
- L1 or L2 loss function
- Which optimizer and what learning rate

Developing the network

Network considerations

- Different feed-forward architectures
- L1 or L2 loss function
- Which optimizer and what learning rate
- Dropout and regularization

Developing the network

Network considerations

- Different feed-forward architectures
- L1 or L2 loss function
- Which optimizer and what learning rate
- Dropout and regularization
- Standardization, scaling and batch normalization

Evaluation

Metrics for evaluating regression model performance

Evaluation

Metrics for evaluating regression model performance
For n samples:

- Mean squared error between prediction y and target t as $\frac{1}{n}\sum_{i=1}^{n}(y_i - t_i)^2$

Evaluation

Metrics for evaluating regression model performance

For n samples:

- Mean squared error between prediction y and target t as $\frac{1}{n}\sum_{i=1}^{n}(y_i - t_i)^2$
- R^2-score, $1 - \frac{SS_{res}}{SS_{tot}}$ where $SS_{res} = \sum_{i=1}^{n}(y_i - t_i)^2$ and $SS_{tot} = \sum_{i=1}^{n}(t_i - \overline{t})^2$.
  Best possible score 1.0.

Table of Contents

Model comparison

- Bigger networks in either depth or width not increasing model performance

Model comparison

- Bigger networks in either depth or width not increasing model performance
- Small dropout fraction improves validation score, no further regularization needed

Model comparison

- Bigger networks in either depth or width not increasing model performance
- Small dropout fraction improves validation score, no further regularization needed
- L2 loss works better, expected since we simulate data and should not have many outliers

Model comparison

- Bigger networks in either depth or width not increasing model performance
- Small dropout fraction improves validation score, no further regularization needed
- L2 loss works better, expected since we simulate data and should not have many outliers
- Batch normalization does not improve, expected since data is already normalized and network not too deep

Model comparison

- Bigger networks in either depth or width not increasing model performance
- Small dropout fraction improves validation score, no further regularization needed
- L2 loss works better, expected since we simulate data and should not have many outliers
- Batch normalization does not improve, expected since data is already normalized and network not too deep
- Scaling outputs to $[0, 1]$ was essential to get any presentable score, since targets very small, around 10^{-7}

Model comparison

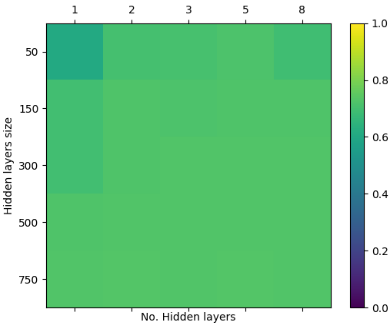


Figure: Heat plot showing R^2-score for hidden layer size vs depth

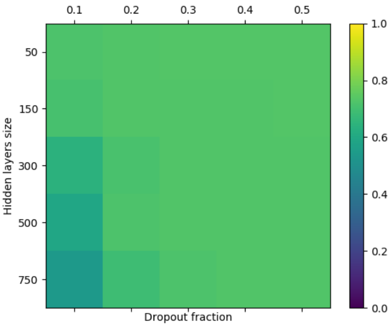


Figure: Heat plot showing R^2-score for hidden layer size vs dropout fraction

Network

Best network configuration

- Simple feedforward network with 3 hidden layers

Network

Best network configuration

- Simple feedforward network with 3 hidden layers
- ReLU activation functions in all but last layer (linear)

Network

Best network configuration

- Simple feedforward network with 3 hidden layers
- ReLU activation functions in all but last layer (linear)
- L2 loss function

Network

Best network configuration

- Simple feedforward network with 3 hidden layers
- ReLU activation functions in all but last layer (linear)
- L2 loss function
- Scaled outputs between $[0, 1]$ since targets very small, around 10^{-7}

Network

Best network configuration

- Simple feedforward network with 3 hidden layers
- ReLU activation functions in all but last layer (linear)
- L2 loss function
- Scaled outputs between $[0, 1]$ since targets very small, around 10^{-7}
- Adam optimizer with learning rate $2.5 * 10^{-4}$

Network

Best network configuration

- Simple feedforward network with 3 hidden layers
- ReLU activation functions in all but last layer (linear)
- L2 loss function
- Scaled outputs between $[0, 1]$ since targets very small, around 10^{-7}
- Adam optimizer with learning rate $2.5 * 10^{-4}$

Test set performance:

- $R^2 = 0.80701$
- $MSE = 1.77545 \times 10^{-14}$

Network

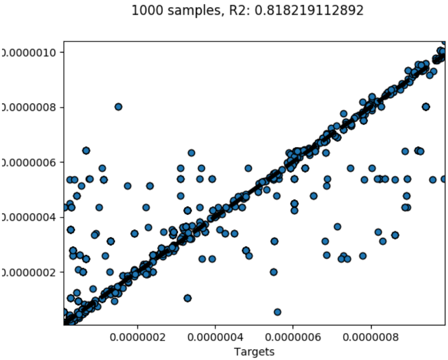


Figure: Predictions vs Targets on 1000 unseen random samples on best performing model

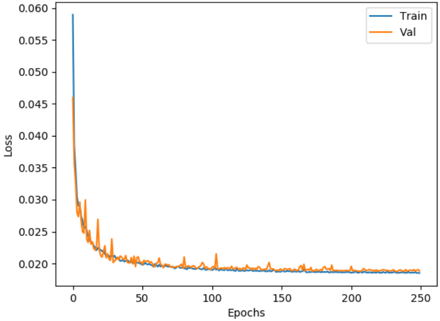


Figure: Loss vs Epochs on best performing model

Table of Contents

Conclusions

- Camino Toolkit at its current state requires a large number of processors to generate big data in reasonable time.

Conclusions

- Camino Toolkit at its current state requires a large number of processors to generate big data in reasonable time.
- Having a regression model with multiple outputs would increase the expressiveness and complexity of the model

Conclusions

- Camino Toolkit at its current state requires a large number of processors to generate big data in reasonable time.
- Having a regression model with multiple outputs would increase the expressiveness and complexity of the model
- Using more complex simulations, i.e not cylinders, would probably be closer to actual diffusion MRI data

Conclusions

- Camino Toolkit at its current state requires a large number of processors to generate big data in reasonable time.
- Having a regression model with multiple outputs would increase the expressiveness and complexity of the model
- Using more complex simulations, i.e not cylinders, would probably be closer to actual diffusion MRI data
- Training and test error very similar even without regularization, may indicate very similar inputs.

Conclusions

- Camino Toolkit at its current state requires a large number of processors to generate big data in reasonable time.
- Having a regression model with multiple outputs would increase the expressiveness and complexity of the model
- Using more complex simulations, i.e not cylinders, would probably be closer to actual diffusion MRI data
- Training and test error very similar even without regularization, may indicate very similar inputs.
- Very easy to learn an "OK" network, very hard to improve on those results.

Conclusions

- Camino Toolkit at its current state requires a large number of processors to generate big data in reasonable time.
- Having a regression model with multiple outputs would increase the expressiveness and complexity of the model
- Using more complex simulations, i.e not cylinders, would probably be closer to actual diffusion MRI data
- Training and test error very similar even without regularization, may indicate very similar inputs.
- Very easy to learn an "OK" network, very hard to improve on those results.
- Difficulty to get high train accuracy may indicate that the problem is ill-posed, i.e inputs are not uniquely mappable to targets.

Thank you for your attention

Questions?