RDM peculiarities, effective receptive fields and surrogate features

Romuald A. Janik

Jagiellonian University
Kraków

RJ 1907.00950 [q-bio.NC]

The competition was especially interesting for me in view of

Bio-inspired Artificial Neural Networks

a 4-year research project at the Jagiellonian University, starting in Fall 2019

involving groups in

- **Machine Learning**
- **Cognitive science**
- **Physics**

we will be hiring **group leaders** for three new groups:

- **(low-level) Neuroscience**
- **Bio Data Science**
- **InfoTech**

some post-doc positions will be available...

any questions: ask me or e-mail me (`romuald.janik@gmail.com`) or see the website bionn.matinf.uj.edu.pl

The competition was especially interesting for me in view of

Bio-inspired Artificial Neural Networks

a 4-year research project at the Jagiellonian University, starting in Fall 2019

involving groups in

Machine Learning
Cognitive science
Physics

we will be hiring **group leaders** for three new groups:

(low-level) Neuroscience
Bio Data Science
InfoTech

some post-doc positions will be available...

any questions: ask me or e-mail me (`romuald.janik@gmail.com`) or see the website bionn.matinf.uj.edu.pl

The competition was especially interesting for me in view of

Bio-inspired Artificial Neural Networks

a 4-year research project at the Jagiellonian University, starting in Fall 2019

involving groups in

Machine Learning
Cognitive science
Physics

we will be hiring **group leaders** for three new groups:

(low-level) Neuroscience
Bio Data Science
InfoTech

some post-doc positions will be available...

any questions: ask me or e-mail me (`romuald.janik@gmail.com`) or see the website bionn.matinf.uj.edu.pl

Main ingredients of the Algonauts Challenge submissions...

RDM peculiarities

Representational Dissimilarity Matrices (RDM) by construction have two rather unexpected and somewhat unwelcome features:

- They can miss a very strong discriminative signal (if correlated)
- They are influenced by irrelevant uninformative features...

$$1 - R(x, y) = 1 - \frac{(x - \langle x \rangle)(y - \langle y \rangle)}{\sigma_x \sigma_y}$$

$$x_i = +1 \qquad y_i = -1$$

RDM will measure only correlation of noise...

RDM peculiarities

Representational Dissimilarity Matrices (RDM) by construction have two rather unexpected and somewhat unwelcome features:

- They can miss a very strong discriminative signal (if correlated)
- They are influenced by irrelevant uninformative features...

$$1 - R(x, y) = 1 - \frac{(x - \langle x \rangle)(y - \langle y \rangle)}{\sigma_x \sigma_y}$$

$$x_i = +1 \qquad y_i = -1$$

RDM will measure only correlation of noise...

RDM peculiarities

Representational Dissimilarity Matrices (RDM) by construction have two rather unexpected and somewhat unwelcome features:

- They can miss a very strong discriminative signal (if correlated)
- They are influenced by irrelevant uninformative features...

$$1 - R(x, y) = 1 - \frac{(x - \langle x \rangle)(y - \langle y \rangle)}{\sigma_x \sigma_y}$$

$$x_i = +1 \qquad y_i = -1$$

RDM will measure only correlation of noise...

RDM peculiarities

Representational Dissimilarity Matrices (RDM) by construction have two rather unexpected and somewhat unwelcome features:

- They can miss a very strong discriminative signal (if correlated)
- They are influenced by irrelevant uninformative features...

$$1 - R(x, y) = 1 - \frac{(x - \langle x \rangle)(y - \langle y \rangle)}{\sigma_x \sigma_y}$$

$x_i = +1 \qquad y_i = -1$

RDM will measure only correlation of noise...

RDM peculiarities

Representational Dissimilarity Matrices (RDM) by construction have two rather unexpected and somewhat unwelcome features:

- They can miss a very strong discriminative signal (if correlated)
- They are influenced by irrelevant uninformative features...

$$1 - R(x, y) = 1 - \frac{(x - \langle x \rangle)(y - \langle y \rangle)}{\sigma_x \sigma_y}$$

$x_i = +1 \qquad y_i = -1$

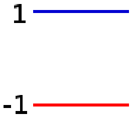
RDM will measure only correlation of noise...

RDM peculiarities

Representational Dissimilarity Matrices (RDM) by construction have two rather unexpected and somewhat unwelcome features:

- They can miss a very strong discriminative signal (if correlated)
- They are influenced by irrelevant uninformative features...

$$1 - R(x, y) = 1 - \frac{(x - \langle x \rangle)(y - \langle y \rangle)}{\sigma_x \sigma_y}$$



$$x_i = +1 \qquad y_i = -1$$

RDM will measure only correlation of noise...

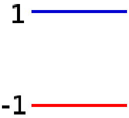
RDM peculiarities

Representational Dissimilarity Matrices (RDM) by construction have two rather unexpected and somewhat unwelcome features:

- They can miss a very strong discriminative signal (if correlated)
- They are influenced by irrelevant uninformative features...

$$1 - R(x, y) = 1 - \frac{(x - \langle x\rangle)(y - \langle y\rangle)}{\sigma_x \sigma_y}$$

$$x_i = +1 + \varepsilon_i \qquad y_i = -1 + \tilde{\varepsilon}_i$$



RDM will measure only correlation of noise...

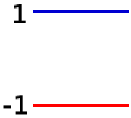
RDM peculiarities

Representational Dissimilarity Matrices (RDM) by construction have two rather unexpected and somewhat unwelcome features:

- They can miss a very strong discriminative signal (if correlated)
- They are influenced by irrelevant uninformative features...

$$1 - R(x, y) = 1 - \frac{(x - \langle x \rangle)(y - \langle y \rangle)}{\sigma_x \sigma_y}$$

$$x_i = +1 + \varepsilon_i \qquad y_i = -1 + \tilde{\varepsilon}_i$$



RDM will measure only correlation of noise...

RDM peculiarities

This behavior (insensitivity to the global signal) can be countered by adding uninformative features...

- This effectively transforms Pearson RDM into *cosine dissimilarity*

$$1 - \frac{x \cdot y}{|x||y|}$$

- This modification significantly increases the scores... (average of NN activations is relevant for describing brain RDM's)
- To some extent, the constant level matters...
- The above suggests another (apart from cosine) possible modification of RDM definition:

$$\langle x \rangle \longrightarrow \text{featurewise average over the dataset}$$

RDM peculiarities

This behavior (insensitivity to the global signal) can be countered by adding uninformative features...

- This effectively transforms Pearson RDM into *cosine dissimilarity*

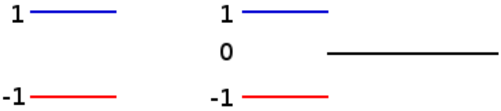
$$1 - \frac{x \cdot y}{|x||y|}$$

- This modification significantly increases the scores...
  (average of NN activations is relevant for describing brain RDM's)
- To some extent, the constant level matters...
- The above suggests another (apart from cosine) possible modification of RDM definition:

$$\langle x \rangle \longrightarrow \text{featurewise average over the dataset}$$

RDM peculiarities

This behavior (insensitivity to the global signal) can be countered by adding uninformative features...



- This effectively transforms Pearson RDM into *cosine dissimilarity*

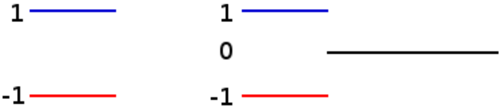
$$1 - \frac{x \cdot y}{|x||y|}$$

- This modification significantly increases the scores...
  (average of NN activations is relevant for describing brain RDM's)
- To some extent, the constant level matters...
- The above suggests another (apart from cosine) possible modification of RDM definition:

$$\langle x \rangle \longrightarrow \text{featurewise average over the dataset}$$

RDM peculiarities

This behavior (insensitivity to the global signal) can be countered by adding uninformative features...



- This effectively transforms Pearson RDM into *cosine dissimilarity*

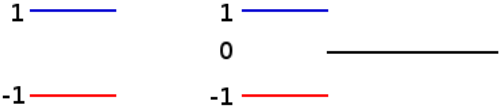
$$1 - \frac{x \cdot y}{|x||y|}$$

- This modification significantly increases the scores... (average of NN activations is relevant for describing brain RDM's)
- To some extent, the constant level matters...
- The above suggests another (apart from cosine) possible modification of RDM definition:

$$\langle x \rangle \longrightarrow \text{featurewise average over the dataset}$$

RDM peculiarities

This behavior (insensitivity to the global signal) can be countered by adding uninformative features...



- This effectively transforms Pearson RDM into *cosine dissimilarity*

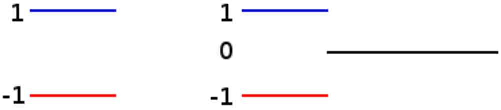
$$1 - \frac{x \cdot y}{|x||y|}$$

- This modification significantly increases the scores...
  (average of NN activations is relevant for describing brain RDM's)
- To some extent, the constant level matters...
- The above suggests another (apart from cosine) possible modification of RDM definition:

$$\langle x \rangle \longrightarrow \text{featurewise average over the dataset}$$

RDM peculiarities

This behavior (insensitivity to the global signal) can be countered by adding uninformative features...



- This effectively transforms Pearson RDM into *cosine dissimilarity*

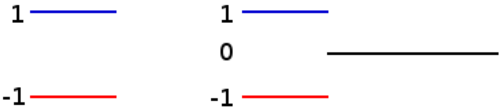
$$1 - \frac{x \cdot y}{|x||y|}$$

- This modification significantly increases the scores...
  (average of NN activations is relevant for describing brain RDM's)
- To some extent, the constant level matters...
- The above suggests another (apart from cosine) possible modification of RDM definition:

$$\langle x \rangle \longrightarrow \text{featurewise average over the dataset}$$

RDM peculiarities

This behavior (insensitivity to the global signal) can be countered by adding uninformative features...



- This effectively transforms Pearson RDM into *cosine dissimilarity*

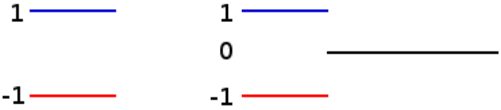
$$1 - \frac{x \cdot y}{|x||y|}$$

- This modification significantly increases the scores...
  (average of NN activations is relevant for describing brain RDM's)
- To some extent, the constant level matters...
- The above suggests another (apart from cosine) possible modification of RDM definition:

$$\langle x \rangle \longrightarrow \text{featurewise average over the dataset}$$

RDM peculiarities

This behavior (insensitivity to the global signal) can be countered by adding uninformative features...



- This effectively transforms Pearson RDM into *cosine dissimilarity*

$$1 - \frac{x \cdot y}{|x||y|}$$

- This modification significantly increases the scores...
  (average of NN activations is relevant for describing brain RDM's)
- To some extent, the constant level matters...
- The above suggests another (apart from cosine) possible modification of RDM definition:

$$\langle x \rangle \longrightarrow \text{featurewise average over the dataset}$$

Effective receptive field

`resnet50`

`block1` $256 \times 56 \times 56$
`block2` $512 \times 28 \times 28$
`block3` $1024 \times 14 \times 14$
`block4` $2048 \times 7 \times 7$

Use `adaptive_max_pool2d` to reduce each layer to $k \times k$

IT: use 2×2
EVC, EARLY, LATE: 5×5

Results **much worse** with average pooling

- NN convolutional features partition the image into various resolutions
- At the same time, features become more higher level...

Question: What is the natural resolution characteristic of brain RDMs?

Effective receptive field

`resnet50`

`block1`	$256 \times 56 \times 56$
`block2`	$512 \times 28 \times 28$
`block3`	$1024 \times 14 \times 14$
`block4`	$2048 \times 7 \times 7$

Use `adaptive_max_pool2d` to reduce each layer to $k \times k$

IT: use 2×2
EVC, EARLY, LATE: 5×5

Results **much worse** with average pooling

- NN convolutional features partition the image into various resolutions
- At the same time, features become more higher level...

Question: What is the natural resolution characteristic of brain RDMs?

Effective receptive field

`resnet50`

`block1` $256 \times 56 \times 56$
`block2` $512 \times 28 \times 28$
`block3` $1024 \times 14 \times 14$
`block4` $2048 \times 7 \times 7$

Use `adaptive_max_pool2d`
to reduce each layer to $k \times k$

IT: use 2×2
EVC, EARLY, LATE: 5×5

Results **much worse** with
average pooling

- NN convolutional features partition the image into various resolutions
- At the same time, features become more higher level...

Question: What is the natural resolution characteristic of brain RDMs?

Effective receptive field

`resnet50`

`block1` $256 \times 56 \times 56$
`block2` $512 \times 28 \times 28$
`block3` $1024 \times 14 \times 14$
`block4` $2048 \times 7 \times 7$

Use `adaptive_max_pool2d` to reduce each layer to $k \times k$

IT: use 2×2
EVC, EARLY, LATE: 5×5

Results **much worse** with average pooling

- NN convolutional features partition the image into various resolutions
- At the same time, features become more higher level...

Question: What is the natural resolution characteristic of brain RDMs?

Effective receptive field

`resnet50`

`block1` $256 \times 56 \times 56$
`block2` $512 \times 28 \times 28$
`block3` $1024 \times 14 \times 14$
`block4` $2048 \times 7 \times 7$

Use `adaptive_max_pool2d` to reduce each layer to $k \times k$

IT: use 2×2
EVC, EARLY, LATE: 5×5

Results **much worse** with average pooling

- NN convolutional features partition the image into various resolutions
- At the same time, features become more higher level...

Question: What is the natural resolution characteristic of brain RDMs?

Effective receptive field

`resnet50`

`block1` $256 \times 56 \times 56$
`block2` $512 \times 28 \times 28$
`block3` $1024 \times 14 \times 14$
`block4` $2048 \times 7 \times 7$

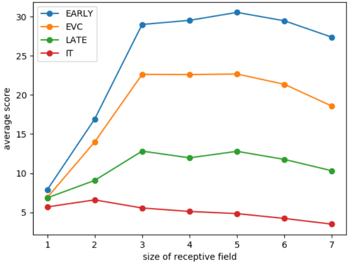
- NN convolutional features partition the image into various resolutions
- At the same time, features become more higher level...

Question: What is the natural resolution characteristic of brain RDMs?

Use `adaptive_max_pool2d` to reduce each layer to $k \times k$

IT: use 2×2
EVC, EARLY, LATE: 5×5

Results **much worse** with average pooling



Effective receptive field

`resnet50`

`block1` $256 \times 56 \times 56$
`block2` $512 \times 28 \times 28$
`block3` $1024 \times 14 \times 14$
`block4` $2048 \times 7 \times 7$

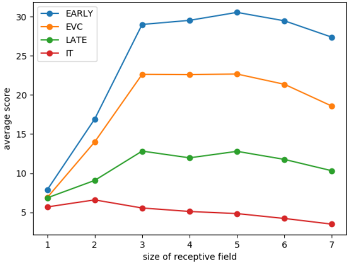
Use `adaptive_max_pool2d` to reduce each layer to $k \times k$

IT: use 2×2
EVC, EARLY, LATE: 5×5

Results **much worse** with average pooling

- NN convolutional features partition the image into various resolutions
- At the same time, features become more higher level...

Question: What is the natural resolution characteristic of brain RDMs?



Effective receptive field

`resnet50`

`block1` $256 \times 56 \times 56$
`block2` $512 \times 28 \times 28$
`block3` $1024 \times 14 \times 14$
`block4` $2048 \times 7 \times 7$

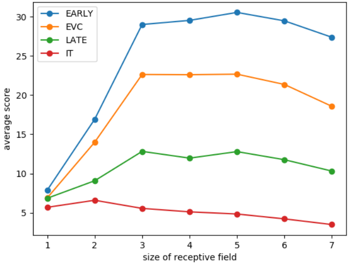
Use `adaptive_max_pool2d` to reduce each layer to $k \times k$

IT: use 2×2
EVC, EARLY, LATE: 5×5

Results **much worse** with average pooling

- NN convolutional features partition the image into various resolutions
- At the same time, features become more higher level...

Question: What is the natural resolution characteristic of brain RDMs?



Feature selection

- RDM is a "global" measure. Features cannot be assessed in isolation... ⟶ first pick some reference set...
- **Erase** or **add** a NN feature ⟶ see how the score changes..
- Try to avoid overfitting...

  (choosing feature weights to maximize score on training dataset does not generalize...)

A For each of the 15 subjects *individually* evaluate the reference score and the modified score (with an added or erased feature). Then take the mean/z-score of the 15 differences.

B Randomly choose 30 subsets of 1/4 images and use these for the reference and modified scores. Take the mean/z-score of the 30 differences.

- For feature pruning use option A on CV test folds as well as on predictions on other dataset...
- For adding features, we used also a modification of B, with 10 different splits into 5 parts, and requiring positivity on both 118 and 92 datasets...

Feature Adding

Feature selection

- RDM is a "global" measure. Features cannot be assessed in isolation... ⟶ first pick some reference set...
- **Erase** or **add** a NN feature ⟶ see how the score changes..
- Try to avoid overfitting...

  (choosing feature weights to maximize score on training dataset does not generalize...)

A For each of the 15 subjects *individually* evaluate the reference score and the modified score (with an added or erased feature). Then take the mean/z-score of the 15 differences.

B Randomly choose 30 subsets of 1/4 images and use these for the reference and modified scores. Take the mean/z-score of the 30 differences.

- For feature pruning use option A on CV test folds as well as on predictions on other dataset...
- For adding features, we used also a modification of B, with 10 different splits into 5 parts, and requiring positivity on both 118 and 92 datasets...

Feature Adding

Feature selection

- RDM is a “global” measure. Features cannot be assessed in isolation... ⟶ first pick some reference set...
- **Erase** or **add** a NN feature ⟶ see how the score changes..
- Try to avoid overfitting...
  (choosing feature weights to maximize score on training dataset does not generalize...)

A For each of the 15 subjects *individually* evaluate the reference score and the modified score (with an added or erased feature). Then take the mean/z-score of the 15 differences.

B Randomly choose 30 subsets of 1/4 images and use these for the reference and modified scores. Take the mean/z-score of the 30 differences.

- For feature pruning use option A on CV test folds as well as on predictions on other dataset...
- For adding features, we used also a modification of B, with 10 different splits into 5 parts, and requiring positivity on both 118 and 92 datasets...

Feature Adding

Feature selection

- RDM is a "global" measure. Features cannot be assessed in isolation... ⟶ first pick some reference set...
- **Erase** or **add** a NN feature ⟶ see how the score changes..
- Try to avoid overfitting...
  (choosing feature weights to maximize score on training dataset does not generalize...)

A For each of the 15 subjects *individually* evaluate the reference score and the modified score (with an added or erased feature). Then take the mean/z-score of the 15 differences.

B Randomly choose 30 subsets of 1/4 images and use these for the reference and modified scores. Take the mean/z-score of the 30 differences.

- For feature pruning use option A on CV test folds as well as on predictions on other dataset...
- For adding features, we used also a modification of B, with 10 different splits into 5 parts, and requiring positivity on both 118 and 92 datasets...

Feature Adding

Feature selection

- RDM is a “global” measure. Features cannot be assessed in isolation... ⟶ first pick some reference set...
- **Erase** or **add** a NN feature ⟶ see how the score changes..
- Try to avoid overfitting...

  (choosing feature weights to maximize score on training dataset does not generalize...)

A For each of the 15 subjects *individually* evaluate the reference score and the modified score (with an added or erased feature). Then take the mean/z-score of the 15 differences.

B Randomly choose 30 subsets of 1/4 images and use these for the reference and modified scores. Take the mean/z-score of the 30 differences.

- For feature pruning use option A on CV test folds as well as on predictions on other dataset...
- For adding features, we used also a modification of B, with 10 different splits into 5 parts, and requiring positivity on both 118 and 92 datasets... **Feature Adding**

Feature selection

- RDM is a "global" measure. Features cannot be assessed in isolation... ⟶ first pick some reference set...
- **Erase** or **add** a NN feature ⟶ see how the score changes..
- Try to avoid overfitting...

  (choosing feature weights to maximize score on training dataset does not generalize...)

A For each of the 15 subjects *individually* evaluate the reference score and the modified score (with an added or erased feature). Then take the mean/z-score of the 15 differences.

B Randomly choose 30 subsets of 1/4 images and use these for the reference and modified scores. Take the mean/z-score of the 30 differences.

- For feature pruning use option A on CV test folds as well as on predictions on other dataset...
- For adding features, we used also a modification of B, with 10 different splits into 5 parts, and requiring positivity on both 118 and 92 datasets...

Feature Adding

Feature selection

- RDM is a "global" measure. Features cannot be assessed in isolation... ⟶ first pick some reference set...
- **Erase** or **add** a NN feature ⟶ see how the score changes..
- Try to avoid overfitting...

  (choosing feature weights to maximize score on training dataset does not generalize...)

A For each of the 15 subjects *individually* evaluate the reference score and the modified score (with an added or erased feature). Then take the mean/z-score of the 15 differences.

B Randomly choose 30 subsets of 1/4 images and use these for the reference and modified scores. Take the mean/z-score of the 30 differences.

- For feature pruning use option A on CV test folds as well as on predictions on other dataset...
- For adding features, we used also a modification of B, with 10 different splits into 5 parts, and requiring positivity on both 118 and 92 datasets...

Feature Adding

Feature selection

- RDM is a "global" measure. Features cannot be assessed in isolation... ⟶ first pick some reference set...
- **Erase** or **add** a NN feature ⟶ see how the score changes..
- Try to avoid overfitting...

  (choosing feature weights to maximize score on training dataset does not generalize...)

A For each of the 15 subjects *individually* evaluate the reference score and the modified score (with an added or erased feature). Then take the mean/z-score of the 15 differences.

B Randomly choose 30 subsets of 1/4 images and use these for the reference and modified scores. Take the mean/z-score of the 30 differences.

- For feature pruning use option A on CV test folds as well as on predictions on other dataset...
- For adding features, we used also a modification of B, with 10 different splits into 5 parts, and requiring positivity on both 118 and 92 datasets...

Feature Adding

Feature selection

- RDM is a "global" measure. Features cannot be assessed in isolation... $\longrightarrow$ first pick some reference set...
- **Erase** or **add** a NN feature $\longrightarrow$ see how the score changes..
- Try to avoid overfitting...

  (choosing feature weights to maximize score on training dataset does not generalize...)

A For each of the 15 subjects *individually* evaluate the reference score and the modified score (with an added or erased feature). Then take the mean/z-score of the 15 differences.

B Randomly choose 30 subsets of 1/4 images and use these for the reference and modified scores. Take the mean/z-score of the 30 differences.

- For feature pruning use option A on CV test folds as well as on predictions on other dataset...
- For adding features, we used also a modification of B, with 10 different splits into 5 parts, and requiring positivity on both 118 and 92 datasets...

Feature Adding

Feature selection

- RDM is a “global” measure. Features cannot be assessed in isolation... ⟶ first pick some reference set...
- **Erase** or **add** a NN feature ⟶ see how the score changes..
- Try to avoid overfitting...

  (choosing feature weights to maximize score on training dataset does not generalize...)

A For each of the 15 subjects *individually* evaluate the reference score and the modified score (with an added or erased feature). Then take the mean/z-score of the 15 differences.

B Randomly choose 30 subsets of 1/4 images and use these for the reference and modified scores. Take the mean/z-score of the 30 differences.

- For feature pruning use option A on CV test folds as well as on predictions on other dataset...
- For adding features, we used also a modification of B, with 10 different splits into 5 parts, and requiring positivity on both 118 and 92 datasets...

Feature Adding

Feature selection

This feature selection procedure (option A) can also be used to study the importance of parts of receptive fields (`maxpool2` of `vgg19` on the 118 image dataset) (positive values bad)

We erase corners in EARLY and EVC...

The score increases also on the test dataset...

Feature selection

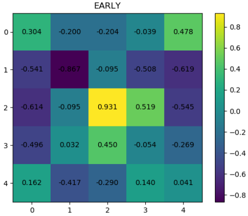
This feature selection procedure (option A) can also be used to study the importance of parts of receptive fields (`maxpool2` of `vgg19` on the 118 image dataset) (positive values bad)

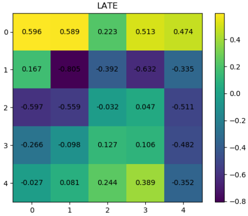
We erase corners in EARLY and EVC...

The score increases also on the test dataset...

Feature selection

This feature selection procedure (option A) can also be used to study the importance of parts of receptive fields (`maxpool2` of `vgg19` on the 118 image dataset) (positive values bad)



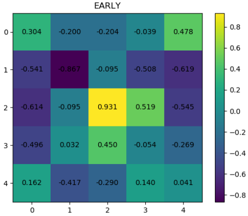


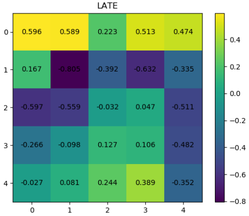
We erase corners in EARLY and EVC...

The score increases also on the test dataset...

Feature selection

This feature selection procedure (option A) can also be used to study the importance of parts of receptive fields (`maxpool2` of `vgg19` on the 118 image dataset) (positive values bad)



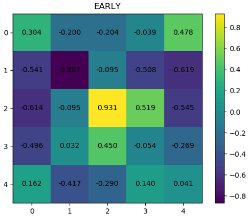


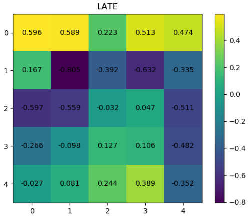
We erase corners in EARLY and EVC...

The score increases also on the test dataset...

Feature selection

This feature selection procedure (option A) can also be used to study the importance of parts of receptive fields (`maxpool2` of `vgg19` on the 118 image dataset) (positive values bad)





We erase corners in EARLY and EVC...

The score increases also on the test dataset...

Solutions for EVC and EARLY

Solutions for EVC and EARLY MEG are very simple...

EVC

1. `block2` of `resnet18` 4.26
2. reduce to 5×5; extend by 0.2 6.41/24.01
3. eliminate $1/4$ of worst features (algorithm B) 25.21
4. eliminate corners; $0.2 \to 0.0$ 26.90/27.57
5. add best features (enhanced $2\times$) 28.29
6. add best features from `maxpool2` of `vgg19` (shrunk $0.5\times$)

Score: 28.40

Erronously adding worst features from other layers instead of **5.** + **6.** gave the best score: 32.68

EARLY

1. `maxpool2` of `vgg19`
2. reduce to 5×5, extend by 0.5
3. eliminate bad features ($z > 0.15$ on either dataset, algorithm A)
4. eliminate corners
5. add best features (enhanced $3\times$)

Score: 46.91

Solutions for EVC and EARLY

Solutions for EVC and EARLY MEG are very simple...

EVC

1. `block2` of `resnet18` 4.26
2. reduce to 5×5; extend by 0.2 6.41/24.01
3. eliminate $1/4$ of worst features (algorithm B) 25.21
4. eliminate corners; $0.2 \rightarrow 0.0$ 26.90/27.57
5. add best features (enhanced $2\times$) 28.29
6. add best features from `maxpool2` of `vgg19` (shrunk $0.5\times$)

Score: 28.40

Erronously adding worst features from other layers instead of **5.** + **6.** gave the best score: 32.68

EARLY

1. `maxpool2` of `vgg19`
2. reduce to 5×5, extend by 0.5
3. eliminate bad features ($z > 0.15$ on either dataset, algorithm A)
4. eliminate corners
5. add best features (enhanced $3\times$)

Score: 46.91

Solutions for EVC and EARLY

Solutions for EVC and EARLY MEG are very simple...

EVC

1. `block2` of `resnet18` 4.26
2. reduce to 5×5; extend by 0.2 6.41/24.01
3. eliminate $1/4$ of worst features (algorithm B) 25.21
4. eliminate corners; $0.2 \to 0.0$ 26.90/27.57
5. add best features (enhanced $2\times$) 28.29
6. add best features from `maxpool2` of `vgg19` (shrunk $0.5\times$)

Score: 28.40

Erronously adding worst features from other layers instead of **5.** + **6.** gave the best score: 32.68

EARLY

1. `maxpool2` of `vgg19`
2. reduce to 5×5, extend by 0.5
3. eliminate bad features ($z > 0.15$ on either dataset, algorithm A)
4. eliminate corners
5. add best features (enhanced $3\times$)

Score: 46.91

Solutions for EVC and EARLY

Solutions for EVC and EARLY MEG are very simple...

EVC

1. `block2` of `resnet18` 4.26
2. reduce to 5×5; extend by 0.2 6.41/24.01
3. eliminate 1/4 of worst features (algorithm B) 25.21
4. eliminate corners; $0.2 \to 0.0$ 26.90/27.57
5. add best features (enhanced $2\times$) 28.29
6. add best features from `maxpool2` of `vgg19` (shrunk $0.5\times$)

Score: 28.40

Erronously adding worst features from other layers instead of **5.** + **6.** gave the best score: 32.68

EARLY

1. `maxpool2` of `vgg19`
2. reduce to 5×5, extend by 0.5
3. eliminate bad features ($z > 0.15$ on either dataset, algorithm A)
4. eliminate corners
5. add best features (enhanced $3\times$)

Score: 46.91

Solutions for EVC and EARLY

Solutions for EVC and EARLY MEG are very simple...

EVC

1. `block2` of `resnet18` 4.26
2. reduce to 5×5; extend by 0.2 6.41/24.01
3. eliminate $1/4$ of worst features (algorithm B) 25.21
4. eliminate corners; $0.2 \to 0.0$ 26.90/27.57
5. add best features (enhanced $2\times$) 28.29
6. add best features from `maxpool2` of `vgg19` (shrunk $0.5\times$)

Score: 28.40

Erronously adding worst features from other layers instead of **5.** + **6.** gave the best score: 32.68

EARLY

1. `maxpool2` of `vgg19`
2. reduce to 5×5, extend by 0.5
3. eliminate bad features ($z > 0.15$ on either dataset, algorithm A)
4. eliminate corners
5. add best features (enhanced $3\times$)

Score: 46.91

Solutions for EVC and EARLY

Solutions for EVC and EARLY MEG are very simple...

EVC

1. block2 of resnet18 4.26
2. reduce to 5×5; extend by 0.2 6.41/24.01
3. eliminate $1/4$ of worst features (algorithm B) 25.21
4. eliminate corners; $0.2 \to 0.0$ 26.90/27.57
5. add best features (enhanced $2\times$) 28.29
6. add best features from maxpool2 of vgg19 (shrunk $0.5\times$)

Score: 28.40

Erronously adding worst features from other layers instead of **5.** + **6.** gave the best score: 32.68

EARLY

1. maxpool2 of vgg19
2. reduce to 5×5, extend by 0.5
3. eliminate bad features ($z > 0.15$ on either dataset, algorithm A)
4. eliminate corners
5. add best features (enhanced $3\times$)

Score: 46.91

Solutions for EVC and EARLY

Solutions for EVC and EARLY MEG are very simple...

EVC

1. `block2` of `resnet18` 4.26
2. reduce to 5×5; extend by 0.2 6.41/24.01
3. eliminate 1/4 of worst features (algorithm B) 25.21
4. eliminate corners; $0.2 \to 0.0$ 26.90/27.57
5. add best features (enhanced $2\times$) 28.29
6. add best features from `maxpool2` of `vgg19` (shrunk $0.5\times$)

Score: 28.40

Erronously adding worst features from other layers instead of **5.** + **6.** gave the best score: 32.68

EARLY

1. `maxpool2` of `vgg19`
2. reduce to 5×5, extend by 0.5
3. eliminate bad features ($z > 0.15$ on either dataset, algorithm A)
4. eliminate corners
5. add best features (enhanced $3\times$)

Score: 46.91

Solutions for EVC and EARLY

Solutions for EVC and EARLY MEG are very simple...

EVC

1. `block2` of `resnet18` 4.26
2. reduce to 5×5; extend by 0.2 6.41/24.01
3. eliminate $1/4$ of worst features (algorithm B) 25.21
4. eliminate corners; $0.2 \to 0.0$ 26.90/27.57
5. add best features (enhanced $2\times$) 28.29
6. add best features from `maxpool2` of `vgg19` (shrunk $0.5\times$)

Score: 28.40

Erronously adding worst features from other layers instead of **5.** + **6.** gave the best score: 32.68

EARLY

1. `maxpool2` of `vgg19`
2. reduce to 5×5, extend by 0.5
3. eliminate bad features ($z > 0.15$ on either dataset, algorithm A)
4. eliminate corners
5. add best features (enhanced $3\times$)

Score: 46.91

Solutions for EVC and EARLY

Solutions for EVC and EARLY MEG are very simple...

EVC

1. `block2` of `resnet18` 4.26
2. reduce to 5×5; extend by 0.2 6.41/24.01
3. eliminate 1/4 of worst features (algorithm B) 25.21
4. eliminate corners; $0.2 \rightarrow 0.0$ 26.90/27.57
5. add best features (enhanced $2\times$) 28.29
6. add best features from `maxpool2` of `vgg19` (shrunk $0.5\times$)

Score: 28.40

Erronously adding worst features from other layers instead of **5.** + **6.** gave the best score: 32.68

EARLY

1. `maxpool2` of `vgg19`
2. reduce to 5×5, extend by 0.5
3. eliminate bad features ($z > 0.15$ on either dataset, algorithm A)
4. eliminate corners
5. add best features (enhanced $3\times$)

Score: 46.91

Solutions for EVC and EARLY

Solutions for EVC and EARLY MEG are very simple...

EVC

1. `block2` of `resnet18` 4.26
2. reduce to 5×5; extend by 0.2 6.41/24.01
3. eliminate $1/4$ of worst features (algorithm B) 25.21
4. eliminate corners; $0.2 \rightarrow 0.0$ 26.90/27.57
5. add best features (enhanced $2\times$) 28.29
6. add best features from `maxpool2` of `vgg19` (shrunk $0.5\times$)

Score: 28.40

Erronously adding worst features from other layers instead of **5.** + **6.** gave the best score: 32.68

EARLY

1. `maxpool2` of `vgg19`
2. reduce to 5×5, extend by 0.5
3. eliminate bad features ($z > 0.15$ on either dataset, algorithm A)
4. eliminate corners
5. add best features (enhanced $3\times$)

Score: 46.91

Solutions for EVC and EARLY

Solutions for EVC and EARLY MEG are very simple...

EVC

1. `block2` of `resnet18` 4.26
2. reduce to 5×5; extend by 0.2 6.41/24.01
3. eliminate $1/4$ of worst features (algorithm B) 25.21
4. eliminate corners; $0.2 \to 0.0$ 26.90/27.57
5. add best features (enhanced $2\times$) 28.29
6. add best features from `maxpool2` of `vgg19` (shrunk $0.5\times$)

Score: 28.40

Erronously adding worst features from other layers instead of **5.** + **6.** gave the best score: 32.68

EARLY

1. `maxpool2` of `vgg19`
2. reduce to 5×5, extend by 0.5
3. eliminate bad features ($z > 0.15$ on either dataset, algorithm A)
4. eliminate corners
5. add best features (enhanced $3\times$)

Score: 46.91

Solutions for EVC and EARLY

Solutions for EVC and EARLY MEG are very simple...

EVC

1. `block2` of `resnet18` 4.26
2. reduce to 5×5; extend by 0.2 6.41/24.01
3. eliminate 1/4 of worst features (algorithm B) 25.21
4. eliminate corners; $0.2 \to 0.0$ 26.90/27.57
5. add best features (enhanced $2\times$) 28.29
6. add best features from `maxpool2` of `vgg19` (shrunk $0.5\times$)

Score: 28.40

Erronously adding worst features from other layers instead of **5.** + **6.** gave the best score: 32.68

EARLY

1. `maxpool2` of `vgg19`
2. reduce to 5×5, extend by 0.5
3. eliminate bad features ($z > 0.15$ on either dataset, algorithm A)
4. eliminate corners
5. add best features (enhanced $3\times$)

Score: 46.91

Solutions for EVC and EARLY

Solutions for EVC and EARLY MEG are very simple...

EVC

1. `block2` of `resnet18` 4.26
2. reduce to 5×5; extend by 0.2 6.41/24.01
3. eliminate 1/4 of worst features (algorithm B) 25.21
4. eliminate corners; $0.2 \to 0.0$ 26.90/27.57
5. add best features (enhanced $2\times$) 28.29
6. add best features from `maxpool2` of `vgg19` (shrunk $0.5\times$)

Score: 28.40

Erronously adding worst features from other layers instead of **5.** + **6.** gave the best score: 32.68

EARLY

1. `maxpool2` of `vgg19`
2. reduce to 5×5, extend by 0.5
3. eliminate bad features ($z > 0.15$ on either dataset, algorithm A)
4. eliminate corners
5. add best features (enhanced $3\times$)

Score: **46.91**

Solutions for EVC and EARLY

Solutions for EVC and EARLY MEG are very simple...

EVC

1. `block2` of `resnet18` 4.26
2. reduce to 5×5; extend by 0.2 6.41/24.01
3. eliminate 1/4 of worst features (algorithm B) 25.21
4. eliminate corners; $0.2 \to 0.0$ 26.90/27.57
5. add best features (enhanced $2\times$) 28.29
6. add best features from `maxpool2` of `vgg19` (shrunk $0.5\times$)

Score: 28.40

Erronously adding worst features from other layers instead of **5.** + **6.** gave the best score: 32.68

EARLY

1. `maxpool2` of `vgg19`
2. reduce to 5×5, extend by 0.5
3. eliminate bad features ($z > 0.15$ on either dataset, algorithm A)
4. eliminate corners
5. add best features (enhanced $3\times$)

Score: 46.91

Solutions for EVC and EARLY

Solutions for EVC and EARLY MEG are very simple...

EVC

1. `block2` of `resnet18` 4.26
2. reduce to 5×5; extend by 0.2 6.41/24.01
3. eliminate $1/4$ of worst features (algorithm B) 25.21
4. eliminate corners; $0.2 \to 0.0$ 26.90/27.57
5. add best features (enhanced $2\times$) 28.29
6. add best features from `maxpool2` of `vgg19` (shrunk $0.5\times$)

Score: 28.40

Erronously adding worst features from other layers instead of **5.** + **6.** gave the best score: 32.68

EARLY

1. `maxpool2` of `vgg19`
2. reduce to 5×5, extend by 0.5
3. eliminate bad features ($z > 0.15$ on either dataset, algorithm A)
4. eliminate corners
5. add best features (enhanced $3\times$)

Score: 46.91

Solutions for EVC and EARLY

Solutions for EVC and EARLY MEG are very simple...

EVC

1. `block2` of `resnet18` 4.26
2. reduce to 5×5; extend by 0.2 6.41/24.01
3. eliminate $1/4$ of worst features (algorithm B) 25.21
4. eliminate corners; $0.2 \rightarrow 0.0$ 26.90/27.57
5. add best features (enhanced $2\times$) 28.29
6. add best features from `maxpool2` of `vgg19` (shrunk $0.5\times$)

Score: 28.40

Erronously adding worst features from other layers instead of **5.** + **6.** gave the best score: 32.68

EARLY

1. `maxpool2` of `vgg19`
2. reduce to 5×5, extend by 0.5
3. eliminate bad features ($z > 0.15$ on either dataset, algorithm A)
4. eliminate corners
5. add best features (enhanced $3\times$)

Score: 46.91

Solutions for EVC and EARLY

Solutions for EVC and EARLY MEG are very simple...

EVC

1. `block2` of `resnet18` 4.26
2. reduce to 5×5; extend by 0.2 6.41/24.01
3. eliminate $1/4$ of worst features (algorithm B) 25.21
4. eliminate corners; $0.2 \rightarrow 0.0$ 26.90/27.57
5. add best features (enhanced $2\times$) 28.29
6. add best features from `maxpool2` of `vgg19` (shrunk $0.5\times$)

Score: 28.40

Erronously adding worst features from other layers instead of **5.** + **6.** gave the best score: 32.68

EARLY

1. `maxpool2` of `vgg19`
2. reduce to 5×5, extend by 0.5
3. eliminate bad features ($z > 0.15$ on either dataset, algorithm A)
4. eliminate corners
5. add best features (enhanced $3\times$)

Score: 46.91

Solutions for EVC and EARLY

Solutions for EVC and EARLY MEG are very simple...

EVC

1. `block2` of `resnet18` 4.26
2. reduce to 5×5; extend by 0.2 6.41/24.01
3. eliminate $1/4$ of worst features (algorithm B) 25.21
4. eliminate corners; $0.2 \to 0.0$ 26.90/27.57
5. add best features (enhanced $2\times$) 28.29
6. add best features from `maxpool2` of `vgg19` (shrunk $0.5\times$)

Score: 28.40

Erronously adding worst features from other layers instead of **5.** + **6.** gave the best score: 32.68

EARLY

1. `maxpool2` of `vgg19`
2. reduce to 5×5, extend by 0.5
3. eliminate bad features ($z > 0.15$ on either dataset, algorithm A)
4. eliminate corners
5. add best features (enhanced $3\times$)

Score: 46.91

Solutions for EVC and EARLY

Solutions for EVC and EARLY MEG are very simple...

EVC

1. `block2` of `resnet18` 4.26
2. reduce to 5×5; extend by 0.2 6.41/24.01
3. eliminate 1/4 of worst features (algorithm B) 25.21
4. eliminate corners; $0.2 \to 0.0$ 26.90/27.57
5. add best features (enhanced $2\times$) 28.29
6. add best features from `maxpool2` of `vgg19` (shrunk $0.5\times$)

Score: 28.40

Erronously adding worst features from other layers instead of **5.** + **6.** gave the best score: 32.68

EARLY

1. `maxpool2` of `vgg19`
2. reduce to 5×5, extend by 0.5
3. eliminate bad features ($z > 0.15$ on either dataset, algorithm A)
4. eliminate corners
5. add best features (enhanced $3\times$)

Score: 46.91

Solutions for IT and LATE – surrogate features

Question: What (abstract) features would reproduce the given brain RDM (averaged across subjects)? (as measured by Spearman's...)

Use **Multidimensional Scaling (MDS)**:

$$\overline{MDS}^{4}_{118\times118} \longrightarrow \mathbb{R}^{118\times10} \qquad \text{repeat with 10 random seeds}$$

using the constructed 100 features gives a score around 77% for the same dataset

General procedure:

1. Fit the resulting 100 features with NN features
   ...fit for each layer individually, then combine fits using ridge regression
2. Drop the bad features (evaluating on CV and/or other dataset)
3. Use the model from **1.** to construct features for the test images... (and drop the bad ones identified in step **2.**)

Solutions for IT and LATE – surrogate features

Question: What (abstract) features would reproduce the given brain RDM (averaged across subjects)? (as measured by Spearman's...)

Use **Multidimensional Scaling (MDS)**:

$$\overline{MDS}^{4}_{118\times 118} \longrightarrow \mathbb{R}^{118\times 10}$$ repeat with 10 random seeds

using the constructed 100 features gives a score around 77% for the same dataset

General procedure:

1. Fit the resulting 100 features with NN features

   ...fit for each layer individually, then combine fits using ridge regression
2. Drop the bad features (evaluating on CV and/or other dataset)
3. Use the model from **1.** to construct features for the test images... (and drop the bad ones identified in step **2.**)

Solutions for IT and LATE – surrogate features

Question: What (abstract) features would reproduce the given brain RDM (averaged across subjects)? (as measured by Spearman's...)

Use **Multidimensional Scaling (MDS)**:

$$\overline{MDS}^{4}_{118\times 118} \longrightarrow \mathbb{R}^{118\times 10}$$ repeat with 10 random seeds

using the constructed 100 features gives a score around 77% for the same dataset

General procedure:

1. Fit the resulting 100 features with NN features
   ...fit for each layer individually, then combine fits using ridge regression
2. Drop the bad features (evaluating on CV and/or other dataset)
3. Use the model from **1.** to construct features for the test images... (and drop the bad ones identified in step **2.**)

Solutions for IT and LATE – surrogate features

Question: What (abstract) features would reproduce the given brain RDM (averaged across subjects)? (as measured by Spearman's...)

Use **Multidimensional Scaling (MDS)**:

$\overline{MDS}^{4}_{118\times118} \longrightarrow \mathbb{R}^{118\times10}$ repeat with 10 random seeds

using the constructed 100 features gives a score around 77% for the same dataset

General procedure:

1. Fit the resulting 100 features with NN features
   ...fit for each layer individually, then combine fits using ridge regression
2. Drop the bad features (evaluating on CV and/or other dataset)
3. Use the model from 1. to construct features for the test images... (and drop the bad ones identified in step 2.)

Solutions for IT and LATE – surrogate features

Question: What (abstract) features would reproduce the given brain RDM (averaged across subjects)? (as measured by Spearman's...)

Use **Multidimensional Scaling (MDS)**:

$\overline{MDS}^{4}_{118\times 118} \longrightarrow \mathbb{R}^{118\times 10}$ repeat with 10 random seeds

using the constructed 100 features gives a score around 77% for the same dataset

General procedure:

1. Fit the resulting 100 features with NN features
   ...fit for each layer individually, then combine fits using ridge regression
2. Drop the bad features (evaluating on CV and/or other dataset)
3. Use the model from **1.** to construct features for the test images... (and drop the bad ones identified in step **2.**)

Solutions for IT and LATE – surrogate features

Question: What (abstract) features would reproduce the given brain RDM (averaged across subjects)? (as measured by Spearman's...)

Use **Multidimensional Scaling (MDS)**:

$$\overline{MDS}^{4}_{118\times 118} \longrightarrow \mathbb{R}^{118\times 10}$$ repeat with 10 random seeds

using the constructed 100 features gives a score around 77% for the same dataset

General procedure:

1. Fit the resulting 100 features with NN features
   ...fit for each layer individually, then combine fits using ridge regression
2. Drop the bad features (evaluating on CV and/or other dataset)
3. Use the model from **1.** to construct features for the test images...
   (and drop the bad ones identified in step **2.**)

Solutions for IT and LATE – surrogate features

Question: What (abstract) features would reproduce the given brain RDM (averaged across subjects)? (as measured by Spearman's...)

Use **Multidimensional Scaling (MDS)**:

$$\overline{MDS}^{4}_{118\times118} \longrightarrow \mathbb{R}^{118\times10}$$

repeat with 10 random seeds

using the constructed 100 features gives a score around 77% for the same dataset

General procedure:

1. Fit the resulting 100 features with NN features

   ...fit for each layer individually, then combine fits using ridge regression

2. Drop the bad features (evaluating on CV and/or other dataset)
3. Use the model from 1. to construct features for the test images... (and drop the bad ones identified in step 2.)

Solutions for IT and LATE – surrogate features

Question: What (abstract) features would reproduce the given brain RDM (averaged across subjects)? (as measured by Spearman's...)

Use **Multidimensional Scaling (MDS)**:

$$\overline{MDS}^{4}_{118\times 118} \longrightarrow \mathbb{R}^{118\times 10}$$

repeat with 10 random seeds

using the constructed 100 features gives a score around 77% for the same dataset

General procedure:

1. Fit the resulting 100 features with NN features

   ...fit for each layer individually, then combine fits using ridge regression

2. Drop the bad features (evaluating on CV and/or other dataset)
3. Use the model from **1.** to construct features for the test images... (and drop the bad ones identified in step **2.**)

Solutions for IT and LATE – surrogate features

Question: What (abstract) features would reproduce the given brain RDM (averaged across subjects)? (as measured by Spearman's...)

Use **Multidimensional Scaling (MDS)**:

$$\overline{MDS}^{4}_{118\times118} \longrightarrow \mathbb{R}^{118\times10}$$

repeat with 10 random seeds

using the constructed 100 features gives a score around 77% for the same dataset

General procedure:

1. Fit the resulting 100 features with NN features

   ...fit for each layer individually, then combine fits using ridge regression

2. Drop the bad features (evaluating on CV and/or other dataset)
3. Use the model from **1.** to construct features for the test images... (and drop the bad ones identified in step **2.**)

Solutions for IT and LATE – surrogate features

Question: What (abstract) features would reproduce the given brain RDM (averaged across subjects)? (as measured by Spearman's...)

Use **Multidimensional Scaling (MDS)**:

$$\overline{MDS}^{4}_{118\times 118} \longrightarrow \mathbb{R}^{118\times 10} \qquad \text{repeat with 10 random seeds}$$

using the constructed 100 features gives a score around 77% for the same dataset

General procedure:

1. Fit the resulting 100 features with NN features

   ...fit for each layer individually, then combine fits using ridge regression
2. Drop the bad features (evaluating on CV and/or other dataset)
3. Use the model from **1**. to construct features for the test images...

   (and drop the bad ones identified in step **2**.)

Solutions for IT and LATE – surrogate features

Question: What (abstract) features would reproduce the given brain RDM (averaged across subjects)? (as measured by Spearman's...)

Use **Multidimensional Scaling (MDS)**:

$$\overline{MDS}^{4}_{118\times 118} \longrightarrow \mathbb{R}^{118\times 10}$$

repeat with 10 random seeds

using the constructed 100 features gives a score around 77% for the same dataset

General procedure:

1. Fit the resulting 100 features with NN features
   ...fit for each layer individually, then combine fits using ridge regression
2. Drop the bad features (evaluating on CV and/or other dataset)
3. Use the model from **1**. to construct features for the test images... (and drop the bad ones identified in step **2**.)

Solutions for IT and LATE – surrogate features

IT

1. Use `resnet50`, convolutional features reduced to 2×2
2. For 118 dataset MDS features: ridge regression; OMP(6)
   For 92 dataset MDS features: OMP(7)
3. Concatenate to get 300 features
4. Prune bad features imposing positivity on 118 dataset
5. Extend with a constant of 1.0
   19.42
6. Add in 75+75 ICA from `block1`, `block3` of `resnet34`

Score: 20.77

LATE

1. Use `resnet50`, convolutional features reduced to 5×5
2. For 118 dataset MDS features: GBR(5) with Huber loss
   For 92 dataset MDS features: GBR(5) with Huber loss
3. Concatenate to get 200 features
4. Prune bad features which are bad (> 0.05) on both datasets
5. Extend with a constant of 1.0

Score: 57.38

Solutions for IT and LATE – surrogate features

IT

1. Use `resnet50`, convolutional features reduced to 2×2
2. For 118 dataset MDS features: ridge regression; OMP(6)
   For 92 dataset MDS features: OMP(7)
3. Concatenate to get 300 features
4. Prune bad features imposing positivity on 118 dataset
5. Extend with a constant of 1.0

   19.42
6. Add in 75+75 ICA from `block1`, `block3` of `resnet34`

Score: 20.77

LATE

1. Use `resnet50`, convolutional features reduced to 5×5
2. For 118 dataset MDS features: GBR(5) with Huber loss
   For 92 dataset MDS features: GBR(5) with Huber loss
3. Concatenate to get 200 features
4. Prune bad features which are bad (> 0.05) on both datasets
5. Extend with a constant of 1.0

Score: 57.38

Solutions for IT and LATE – surrogate features

IT

1. Use `resnet50`, convolutional features reduced to 2×2
2. For 118 dataset MDS features:
   ridge regression; OMP(6)
   For 92 dataset MDS features:
   OMP(7)
3. Concatenate to get 300 features
4. Prune bad features imposing positivity on 118 dataset
5. Extend with a constant of 1.0
   19.42
6. Add in 75+75 ICA from `block1`, `block3` of `resnet34`

Score: **20.77**

LATE

1. Use `resnet50`, convolutional features reduced to 5×5
2. For 118 dataset MDS features:
   GBR(5) with Huber loss
   For 92 dataset MDS features:
   GBR(5) with Huber loss
3. Concatenate to get 200 features
4. Prune bad features which are bad (> 0.05) on both datasets
5. Extend with a constant of 1.0

Score: **57.38**

Solutions for IT and LATE – surrogate features

IT

1. Use `resnet50`, convolutional features reduced to 2×2
2. For 118 dataset MDS features: ridge regression; OMP(6)
   For 92 dataset MDS features: OMP(7)
3. Concatenate to get 300 features
4. Prune bad features imposing positivity on 118 dataset
5. Extend with a constant of 1.0

19.42

6. Add in 75+75 ICA from `block1`, `block3` of `resnet34`

Score: **20.77**

LATE

1. Use `resnet50`, convolutional features reduced to 5×5
2. For 118 dataset MDS features: GBR(5) with Huber loss
   For 92 dataset MDS features: GBR(5) with Huber loss
3. Concatenate to get 200 features
4. Prune bad features which are bad (> 0.05) on both datasets
5. Extend with a constant of 1.0

Score: **57.38**

Solutions for IT and LATE – surrogate features

IT

1. Use `resnet50`, convolutional features reduced to 2×2
2. For 118 dataset MDS features:
   ridge regression; OMP(6)
   For 92 dataset MDS features:
   OMP(7)
3. Concatenate to get 300 features
4. Prune bad features imposing positivity on 118 dataset
5. Extend with a constant of 1.0
   19.42
6. Add in 75+75 ICA from `block1`, `block3` of `resnet34`

Score: 20.77

LATE

1. Use `resnet50`, convolutional features reduced to 5×5
2. For 118 dataset MDS features:
   GBR(5) with Huber loss
   For 92 dataset MDS features:
   GBR(5) with Huber loss
3. Concatenate to get 200 features
4. Prune bad features which are bad (> 0.05) on both datasets
5. Extend with a constant of 1.0

Score: 57.38

Solutions for IT and LATE – surrogate features

IT

1. Use `resnet50`, convolutional features reduced to 2×2
2. For 118 dataset MDS features: ridge regression; OMP(6)
   For 92 dataset MDS features: OMP(7)
3. Concatenate to get 300 features
4. Prune bad features imposing positivity on 118 dataset
5. Extend with a constant of 1.0
   19.42
6. Add in 75+75 ICA from `block1`, `block3` of `resnet34`

Score: 20.77

LATE

1. Use `resnet50`, convolutional features reduced to 5×5
2. For 118 dataset MDS features: GBR(5) with Huber loss
   For 92 dataset MDS features: GBR(5) with Huber loss
3. Concatenate to get 200 features
4. Prune bad features which are bad (> 0.05) on both datasets
5. Extend with a constant of 1.0

Score: 57.38

Solutions for IT and LATE – surrogate features

IT

1. Use `resnet50`, convolutional features reduced to 2×2
2. For 118 dataset MDS features: ridge regression; OMP(6)
   For 92 dataset MDS features: OMP(7)
3. Concatenate to get 300 features
4. Prune bad features imposing positivity on 118 dataset
5. Extend with a constant of 1.0

   19.42

6. Add in 75+75 ICA from `block1`, `block3` of `resnet34`

Score: 20.77

LATE

1. Use `resnet50`, convolutional features reduced to 5×5
2. For 118 dataset MDS features: GBR(5) with Huber loss
   For 92 dataset MDS features: GBR(5) with Huber loss
3. Concatenate to get 200 features
4. Prune bad features which are bad (> 0.05) on both datasets
5. Extend with a constant of 1.0

Score: 57.38

Solutions for IT and LATE – surrogate features

IT

1. Use `resnet50`, convolutional features reduced to 2×2
2. For 118 dataset MDS features: ridge regression; OMP(6)
   For 92 dataset MDS features: OMP(7)
3. Concatenate to get 300 features
4. Prune bad features imposing positivity on 118 dataset
5. Extend with a constant of 1.0
   19.42
6. Add in 75+75 ICA from `block1`, `block3` of `resnet34`

Score: 20.77

LATE

1. Use `resnet50`, convolutional features reduced to 5×5
2. For 118 dataset MDS features: GBR(5) with Huber loss
   For 92 dataset MDS features: GBR(5) with Huber loss
3. Concatenate to get 200 features
4. Prune bad features which are bad (> 0.05) on both datasets
5. Extend with a constant of 1.0

Score: 57.38

Solutions for IT and LATE – surrogate features

IT

1. Use `resnet50`, convolutional features reduced to 2×2
2. For 118 dataset MDS features: ridge regression; OMP(6)
   For 92 dataset MDS features: OMP(7)
3. Concatenate to get 300 features
4. Prune bad features imposing positivity on 118 dataset
5. Extend with a constant of 1.0
   19.42
6. Add in 75+75 ICA from `block1`, `block3` of `resnet34`

Score: **20.77**

LATE

1. Use `resnet50`, convolutional features reduced to 5×5
2. For 118 dataset MDS features: GBR(5) with Huber loss
   For 92 dataset MDS features: GBR(5) with Huber loss
3. Concatenate to get 200 features
4. Prune bad features which are bad (> 0.05) on both datasets
5. Extend with a constant of 1.0

Score: **57.38**

Solutions for IT and LATE – surrogate features

IT

1. Use `resnet50`, convolutional features reduced to 2×2
2. For 118 dataset MDS features:
   ridge regression; OMP(6)
   For 92 dataset MDS features:
   OMP(7)
3. Concatenate to get 300 features
4. Prune bad features imposing positivity on 118 dataset
5. Extend with a constant of 1.0

   19.42
6. Add in 75+75 ICA from `block1`, `block3` of `resnet34`

Score: 20.77

LATE

1. Use `resnet50`, convolutional features reduced to 5×5
2. For 118 dataset MDS features:
   GBR(5) with Huber loss
   For 92 dataset MDS features:
   GBR(5) with Huber loss
3. Concatenate to get 200 features
4. Prune bad features which are bad (> 0.05) on both datasets
5. Extend with a constant of 1.0

Score: 57.38

Solutions for IT and LATE – surrogate features

IT

1. Use `resnet50`, convolutional features reduced to 2×2
2. For 118 dataset MDS features: ridge regression; OMP(6)
   For 92 dataset MDS features: OMP(7)
3. Concatenate to get 300 features
4. Prune bad features imposing positivity on 118 dataset
5. Extend with a constant of 1.0

   19.42
6. Add in 75+75 ICA from `block1`, `block3` of `resnet34`

Score: 20.77

LATE

1. Use `resnet50`, convolutional features reduced to 5×5
2. For 118 dataset MDS features: GBR(5) with Huber loss
   For 92 dataset MDS features: GBR(5) with Huber loss
3. Concatenate to get 200 features
4. Prune bad features which are bad (> 0.05) on both datasets
5. Extend with a constant of 1.0

Score: 57.38

Solutions for IT and LATE – surrogate features

IT

1. Use `resnet50`, convolutional features reduced to 2×2
2. For 118 dataset MDS features: ridge regression; OMP(6)
   For 92 dataset MDS features: OMP(7)
3. Concatenate to get 300 features
4. Prune bad features imposing positivity on 118 dataset
5. Extend with a constant of 1.0
   19.42
6. Add in 75+75 ICA from `block1`, `block3` of `resnet34`

Score: 20.77

LATE

1. Use `resnet50`, convolutional features reduced to 5×5
2. For 118 dataset MDS features: GBR(5) with Huber loss
   For 92 dataset MDS features: GBR(5) with Huber loss
3. Concatenate to get 200 features
4. Prune bad features which are bad (> 0.05) on both datasets
5. Extend with a constant of 1.0

Score: 57.38

Solutions for IT and LATE – surrogate features

IT

1. Use `resnet50`, convolutional features reduced to 2×2
2. For 118 dataset MDS features: ridge regression; OMP(6)
   For 92 dataset MDS features: OMP(7)
3. Concatenate to get 300 features
4. Prune bad features imposing positivity on 118 dataset
5. Extend with a constant of 1.0
   19.42
6. Add in 75+75 ICA from `block1`, `block3` of `resnet34`

Score: 20.77

LATE

1. Use `resnet50`, convolutional features reduced to 5×5
2. For 118 dataset MDS features: GBR(5) with Huber loss
   For 92 dataset MDS features: GBR(5) with Huber loss
3. Concatenate to get 200 features
4. Prune bad features which are bad (> 0.05) on both datasets
5. Extend with a constant of 1.0

Score: 57.38

Solutions for IT and LATE – surrogate features

IT

1. Use `resnet50`, convolutional features reduced to 2×2
2. For 118 dataset MDS features:
   ridge regression; OMP(6)
   For 92 dataset MDS features:
   OMP(7)
3. Concatenate to get 300 features
4. Prune bad features imposing positivity on 118 dataset
5. Extend with a constant of 1.0
   19.42
6. Add in 75+75 ICA from `block1`, `block3` of `resnet34`

Score: 20.77

LATE

1. Use `resnet50`, convolutional features reduced to 5×5
2. For 118 dataset MDS features:
   GBR(5) with Huber loss
   For 92 dataset MDS features:
   GBR(5) with Huber loss
3. Concatenate to get 200 features
4. Prune bad features which are bad (> 0.05) on both datasets
5. Extend with a constant of 1.0

Score: 57.38

Solutions for IT and LATE – surrogate features

IT

1. Use `resnet50`, convolutional features reduced to 2×2
2. For 118 dataset MDS features: ridge regression; OMP(6)
   For 92 dataset MDS features: OMP(7)
3. Concatenate to get 300 features
4. Prune bad features imposing positivity on 118 dataset
5. Extend with a constant of 1.0

   19.42

6. Add in 75+75 ICA from `block1`, `block3` of `resnet34`

Score: 20.77

LATE

1. Use `resnet50`, convolutional features reduced to 5×5
2. For 118 dataset MDS features: GBR(5) with Huber loss
   For 92 dataset MDS features: GBR(5) with Huber loss
3. Concatenate to get 200 features
4. Prune bad features which are bad (> 0.05) on both datasets
5. Extend with a constant of 1.0

Score: 57.38

Solutions for IT and LATE – surrogate features

IT

1. Use `resnet50`, convolutional features reduced to 2×2
2. For 118 dataset MDS features:
   ridge regression; OMP(6)
   For 92 dataset MDS features:
   OMP(7)
3. Concatenate to get 300 features
4. Prune bad features imposing positivity on 118 dataset
5. Extend with a constant of 1.0
   19.42
6. Add in 75+75 ICA from `block1`, `block3` of `resnet34`

Score: 20.77

LATE

1. Use `resnet50`, convolutional features reduced to 5×5
2. For 118 dataset MDS features:
   GBR(5) with Huber loss
   For 92 dataset MDS features:
   GBR(5) with Huber loss
3. Concatenate to get 200 features
4. Prune bad features which are bad (> 0.05) on both datasets
5. Extend with a constant of 1.0

Score: 57.38

Summary and outlook

- Key difficulty: overfitting
  - lots of NN features versus small number of images
  - The three datasets were quite distinct...
- Sometimes CV, as well as assessment of feature importance, was not reliable
- Try to stick to simple models...
- The receptive field reductions to 5×5 (or 2×2 for IT) seemed to be quite robust for all datasets.
- "second level" (`block2` or `maxpool2`) NN features seem to be a good staring point...
- Max-pooling much better than average-pooling...

- Perhaps it would be better to modify the definition of RDM to eliminate the peculiarities mentioned here...
- Instead of MDS, one can generate features (embedding) to approximate RDM minimizing the mean squared error...
- Of course, instead of surrogate features one could model parts of the fMRI signal directly...

Summary and outlook

- Key difficulty: overfitting
  - lots of NN features versus small number of images
  - The three datasets were quite distinct...
- Sometimes CV, as well as assessment of feature importance, was not reliable
- Try to stick to simple models...
- The receptive field reductions to 5×5 (or 2×2 for IT) seemed to be quite robust for all datasets.
- "second level" (`block2` or `maxpool2`) NN features seem to be a good staring point...
- Max-pooling much better than average-pooling...

- Perhaps it would be better to modify the definition of RDM to eliminate the peculiarities mentioned here...
- Instead of MDS, one can generate features (embedding) to approximate RDM minimizing the mean squared error...
- Of course, instead of surrogate features one could model parts of the fMRI signal directly...

Summary and outlook

- Key difficulty: overfitting
  - lots of NN features versus small number of images
  - The three datasets were quite distinct...
- Sometimes CV, as well as assessment of feature importance, was not reliable
- Try to stick to simple models...
- The receptive field reductions to 5×5 (or 2×2 for IT) seemed to be quite robust for all datasets.
- "second level" (`block2` or `maxpool2`) NN features seem to be a good staring point...
- Max-pooling much better than average-pooling...

- Perhaps it would be better to modify the definition of RDM to eliminate the peculiarities mentioned here...
- Instead of MDS, one can generate features (embedding) to approximate RDM minimizing the mean squared error...
- Of course, instead of surrogate features one could model parts of the fMRI signal directly...

Summary and outlook

- Key difficulty: overfitting
  - lots of NN features versus small number of images
  - The three datasets were quite distinct...
- Sometimes CV, as well as assessment of feature importance, was not reliable
- Try to stick to simple models...
- The receptive field reductions to 5×5 (or 2×2 for IT) seemed to be quite robust for all datasets.
- "second level" (`block2` or `maxpool2`) NN features seem to be a good staring point...
- Max-pooling much better than average-pooling...

- Perhaps it would be better to modify the definition of RDM to eliminate the peculiarities mentioned here...
- Instead of MDS, one can generate features (embedding) to approximate RDM minimizing the mean squared error...
- Of course, instead of surrogate features one could model parts of the fMRI signal directly...

Summary and outlook

- Key difficulty: overfitting
  - lots of NN features versus small number of images
  - The three datasets were quite distinct...
- Sometimes CV, as well as assessment of feature importance, was not reliable
- Try to stick to simple models...
- The receptive field reductions to 5×5 (or 2×2 for IT) seemed to be quite robust for all datasets.
- "second level" (`block2` or `maxpool2`) NN features seem to be a good staring point...
- Max-pooling much better than average-pooling...

- Perhaps it would be better to modify the definition of RDM to eliminate the peculiarities mentioned here...
- Instead of MDS, one can generate features (embedding) to approximate RDM minimizing the mean squared error...
- Of course, instead of surrogate features one could model parts of the fMRI signal directly...

Summary and outlook

- Key difficulty: overfitting
  - lots of NN features versus small number of images
  - The three datasets were quite distinct...
- Sometimes CV, as well as assessment of feature importance, was not reliable
- Try to stick to simple models...
- The receptive field reductions to 5×5 (or 2×2 for IT) seemed to be quite robust for all datasets.
- ”second level” (`block2` or `maxpool2`) NN features seem to be a good staring point...
- Max-pooling much better than average-pooling...

- Perhaps it would be better to modify the definition of RDM to eliminate the peculiarities mentioned here...
- Instead of MDS, one can generate features (embedding) to approximate RDM minimizing the mean squared error...
- Of course, instead of surrogate features one could model parts of the fMRI signal directly...

Summary and outlook

- Key difficulty: overfitting
  - lots of NN features versus small number of images
  - The three datasets were quite distinct...
- Sometimes CV, as well as assessment of feature importance, was not reliable
- Try to stick to simple models...
- The receptive field reductions to 5×5 (or 2×2 for IT) seemed to be quite robust for all datasets.
- "second level" (`block2` or `maxpool2`) NN features seem to be a good staring point...
- Max-pooling much better than average-pooling...

- Perhaps it would be better to modify the definition of RDM to eliminate the peculiarities mentioned here...
- Instead of MDS, one can generate features (embedding) to approximate RDM minimizing the mean squared error...
- Of course, instead of surrogate features one could model parts of the fMRI signal directly...

Summary and outlook

- Key difficulty: overfitting
  - lots of NN features versus small number of images
  - The three datasets were quite distinct...
- Sometimes CV, as well as assessment of feature importance, was not reliable
- Try to stick to simple models...
- The receptive field reductions to 5×5 (or 2×2 for IT) seemed to be quite robust for all datasets.
- "second level" (`block2` or `maxpool2`) NN features seem to be a good staring point...
- Max-pooling much better than average-pooling...

- Perhaps it would be better to modify the definition of RDM to eliminate the peculiarities mentioned here...
- Instead of MDS, one can generate features (embedding) to approximate RDM minimizing the mean squared error...
- Of course, instead of surrogate features one could model parts of the fMRI signal directly...

Summary and outlook

- Key difficulty: overfitting
  - lots of NN features versus small number of images
  - The three datasets were quite distinct...
- Sometimes CV, as well as assessment of feature importance, was not reliable
- Try to stick to simple models...
- The receptive field reductions to 5×5 (or 2×2 for IT) seemed to be quite robust for all datasets.
- "second level" (`block2` or `maxpool2`) NN features seem to be a good staring point...
- Max-pooling much better than average-pooling...

- Perhaps it would be better to modify the definition of RDM to eliminate the peculiarities mentioned here...
- Instead of MDS, one can generate features (embedding) to approximate RDM minimizing the mean squared error...
- Of course, instead of surrogate features one could model parts of the fMRI signal directly...

Summary and outlook

- Key difficulty: overfitting
  - lots of NN features versus small number of images
  - The three datasets were quite distinct...
- Sometimes CV, as well as assessment of feature importance, was not reliable
- Try to stick to simple models...
- The receptive field reductions to 5×5 (or 2×2 for IT) seemed to be quite robust for all datasets.
- "second level" (`block2` or `maxpool2`) NN features seem to be a good staring point...
- Max-pooling much better than average-pooling...

- Perhaps it would be better to modify the definition of RDM to eliminate the peculiarities mentioned here...
- Instead of MDS, one can generate features (embedding) to approximate RDM minimizing the mean squared error...
- Of course, instead of surrogate features one could model parts of the fMRI signal directly...

Summary and outlook

- Key difficulty: overfitting
  - lots of NN features versus small number of images
  - The three datasets were quite distinct...
- Sometimes CV, as well as assessment of feature importance, was not reliable
- Try to stick to simple models...
- The receptive field reductions to 5×5 (or 2×2 for IT) seemed to be quite robust for all datasets.
- "second level" (`block2` or `maxpool2`) NN features seem to be a good staring point...
- Max-pooling much better than average-pooling...

- Perhaps it would be better to modify the definition of RDM to eliminate the peculiarities mentioned here...
- Instead of MDS, one can generate features (embedding) to approximate RDM minimizing the mean squared error...
- Of course, instead of surrogate features one could model parts of the fMRI signal directly...

Summary and outlook

- Key difficulty: overfitting
  - lots of NN features versus small number of images
  - The three datasets were quite distinct...
- Sometimes CV, as well as assessment of feature importance, was not reliable
- Try to stick to simple models...
- The receptive field reductions to 5×5 (or 2×2 for IT) seemed to be quite robust for all datasets.
- "second level" (`block2` or `maxpool2`) NN features seem to be a good staring point...
- Max-pooling much better than average-pooling...

- Perhaps it would be better to modify the definition of RDM to eliminate the peculiarities mentioned here...
- Instead of MDS, one can generate features (embedding) to approximate RDM minimizing the mean squared error...
- Of course, instead of surrogate features one could model parts of the fMRI signal directly...