Shortcut Learning in Deep Neural Networks

Robert Geirhos, Jorn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, & Felix Wichmann

Published in *Nature Machine Intelligence*, November 2020
Already cited 100+ times

Presented by Caroline Hixon

Outline

I. Introduction to Shortcut Learning

II. Shortcut leaning in biological neural networks

III. Shortcuts defined: a taxonomy of decision rules

IV. Shortcuts: where do they come from?

V. Shortcut learning across deep learning

VI. Diagnosing and understanding shortcut learning

VII. Beyond shortcut learning

VIII. What other research is being done?

I. Introduction

- Still early in field of deep learning

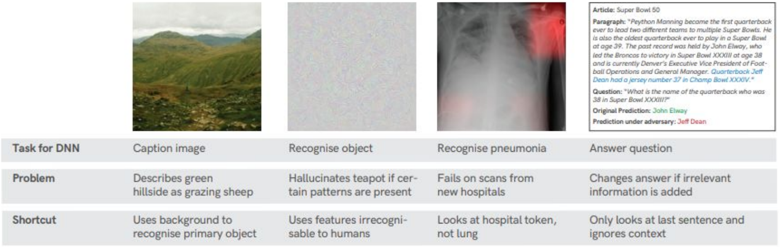
- Success often overshadows lack of understanding and limitations
  - Go, Poker, detecting cancer from X-ray scans

- Deep learning commonplace in our lives and society
  - Possibility to negatively impact society: autonomous vehicles, job applications, cancer screenings

- We need to know when does deep learning *work*, when does it *fail*, and *why*?

I. Introduction

- Currently faced with large number of failure cases



Article: Super Bowl 50

Paragraph: *"Peython Manning became the first quarterback ever to lead two different teams to multiple Super Bowls. He is also the oldest quarterback ever to play in a Super Bowl at age 39. The past record was held by John Elway, who led the Broncos to victory in Super Bowl XXXIII at age 38 and is currently Denver's Executive Vice President of Football Operations and General Manager. Quarterback Jeff Dean had a jersey number 37 in Champ Bowl XXXIV."*

Question: *"What is the name of the quarterback who was 38 in Super Bowl XXXIII?"*

Original Prediction: John Elway

Prediction under adversary: Jeff Dean

Task for DNN	Caption image	Recognise object	Recognise pneumonia	Answer question
Problem	Describes green hillside as grazing sheep	Hallucinates teapot if certain patterns are present	Fails on scans from new hospitals	Changes answer if irrelevant information is added
Shortcut	Uses background to recognise primary object	Uses features irrecognisable to humans	Looks at hospital token, not lung	Only looks at last sentence and ignores context

Figure 1. Deep neural networks often solve problems by taking shortcuts instead of learning the intended solution, leading to a lack of generalisation and unintuitive failures. This pattern can be observed in many real-world applications.

- What do we know about these failures? → not independent, but follow unintended *shortcut strategies* that fail under slightly different circumstances

- Shortcuts revealed when intended solution doesn't match learned solution (sheep)

I. Introduction

Motivations and contributions of paper

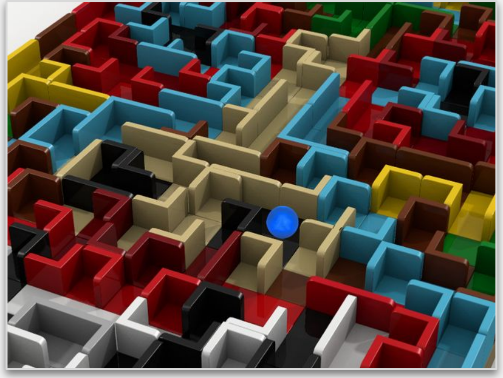
(1) Present unifying view of shortcuts

(2) Identify the common themes

(3) Explain approaches in theory and practice

II. Shortcut learning in biological neural networks

(1) Comparative Psychology: unintended cue learning



- Intended solution: navigate maze based on *color*
- Learned solution: discriminate colors by *odor* of paint

- Animals prone to *unintended cue learning*

II. Shortcut learning in biological neural networks

(2) Education: surface learning

- Intended solution: perform well on test by *understanding* information
- Learned solution: *reproductive* learning that uses simple discrimination

- Gives appearance of good performance, but will fail under different testing circumstances

III. Shortcuts defined: a taxonomy of decision rules

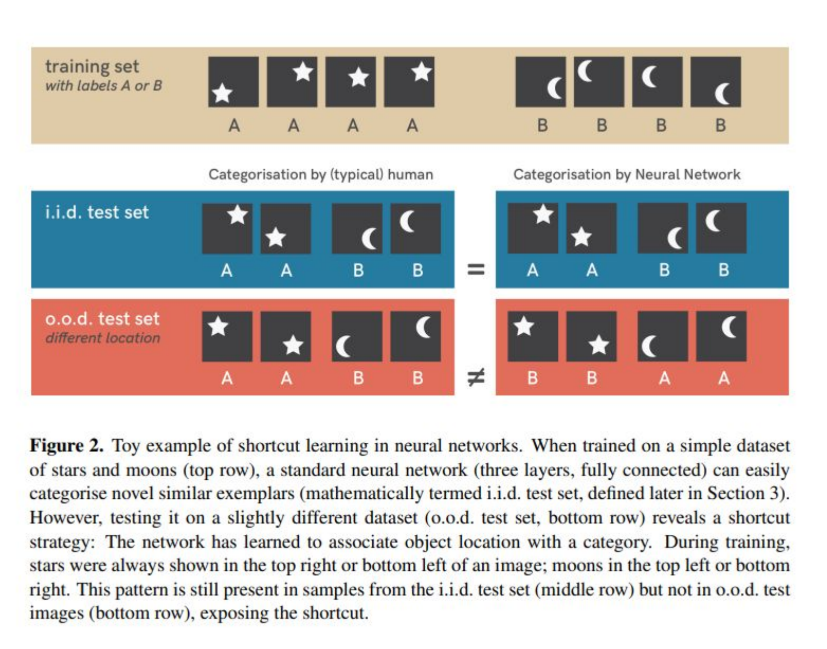


Figure 2. Toy example of shortcut learning in neural networks. When trained on a simple dataset of stars and moons (top row), a standard neural network (three layers, fully connected) can easily categorise novel similar exemplars (mathematically termed i.i.d. test set, defined later in Section 3). However, testing it on a slightly different dataset (o.o.d. test set, bottom row) reveals a shortcut strategy: The network has learned to associate object location with a category. During training, stars were always shown in the top right or bottom left of an image; moons in the top left or bottom right. This pattern is still present in samples from the i.i.d. test set (middle row) but not in o.o.d. test images (bottom row), exposing the shortcut.

Toy example

- Intended solution: identify by shape
- Actual solution: identify by location

Neural networks use decision rules that define a relationship between input and output

- Shortcuts are particular group of decision rules

III. Shortcuts defined: a taxonomy of decision rules

(1) All possible decision rules, including non-solutions

- Ex: if image has white pixel → star
- *Uninformative features* give models poor performance

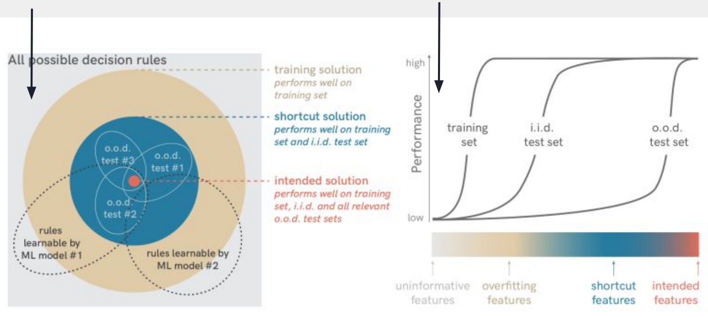
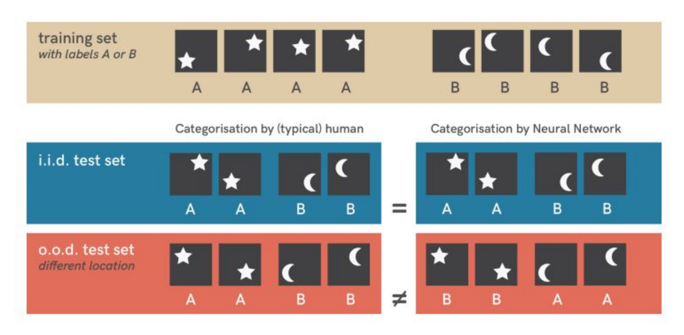

All possible decision rules
training solution
performs well on training set
shortcut solution
performs well on training set and i.i.d. test set
intended solution
performs well on training set, i.i.d. and all relevant o.o.d. test sets
o.o.d. test #3
o.o.d. test #1
o.o.d. test #2
rules learnable by ML model #1
rules learnable by ML model #2
high
Performance
low
training set
i.i.d. test set
o.o.d. test set
uninformative features
overfitting features
shortcut features
intended features

Figure 3. Taxonomy of decision rules. Among the set of all possible rules, only some solve the training data. Among the solutions that solve the training data, only some generalise to an i.i.d. test set. Among those solutions, shortcuts fail to generalise to different data (o.o.d. test sets), but the intended solution does generalise.



III. Shortcuts defined: a taxonomy of decision rules

(2) Training solutions, including overfitting solutions

- Randomized train/test → independent and identically distributed (i.i.d.) test data
- *Overfitting features* give model high performance on training data, but not test

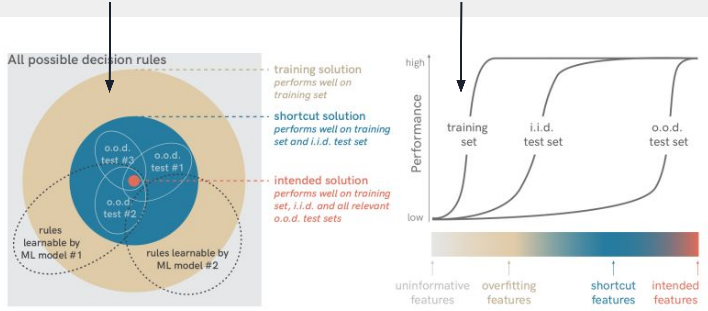
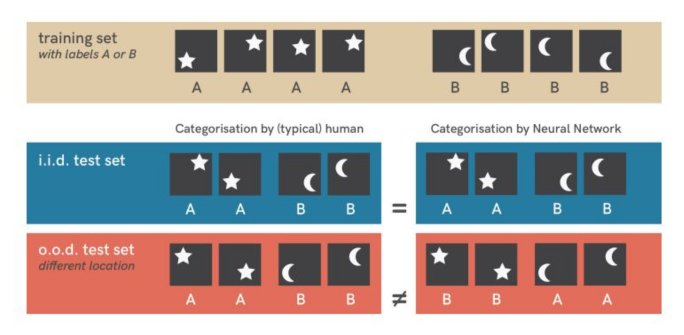

All possible decision rules
training solution
performs well on training set
shortcut solution
performs well on training set and i.i.d. test set
intended solution
performs well on training set, i.i.d. and all relevant o.o.d. test sets
o.o.d. test #3
o.o.d. test #1
o.o.d. test #2
rules learnable by ML model #1
rules learnable by ML model #2
high
Performance
low
training set
i.i.d. test set
o.o.d. test set
uninformative features
overfitting features
shortcut features
intended features

Figure 3. Taxonomy of decision rules. Among the set of all possible rules, only some solve the training data. Among the solutions that solve the training data, only some generalise to an i.i.d. test set. Among those solutions, shortcuts fail to generalise to different data (o.o.d. test sets), but the intended solution does generalise.



III. Shortcuts defined: a taxonomy of decision rules

(3) i.i.d. test solutions, including shortcuts

- Only testing with i.i.d means cannot identify shortcuts
- Out of distribution data (o.o.d.) is systematically different than i.i.d.
- *Shortcut features* revealed when perform well on training and i.i.d., but not o.o.d.

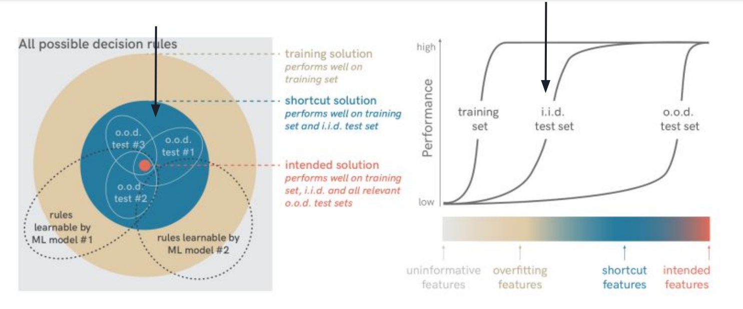
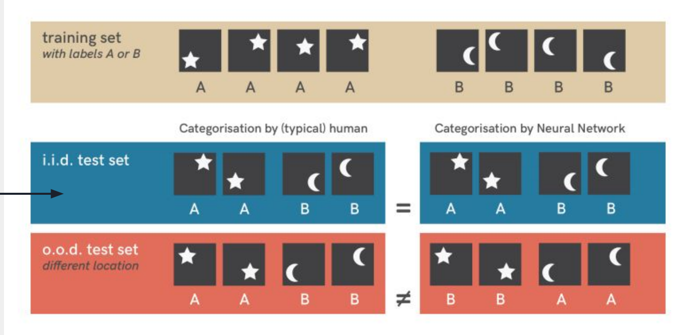

All possible decision rules
training solution
performs well on training set
shortcut solution
performs well on training set and i.i.d. test set
intended solution
performs well on training set, i.i.d. and all relevant o.o.d. test sets
o.o.d. test #3
o.o.d. test #1
o.o.d. test #2
rules learnable by ML model #1
rules learnable by ML model #2
high
Performance
low
training set
i.i.d. test set
o.o.d. test set
uninformative features
overfitting features
shortcut features
intended features

Figure 3. Taxonomy of decision rules. Among the set of all possible rules, only some solve the training data. Among the solutions that solve the training data, only some generalise to an i.i.d. test set. Among those solutions, shortcuts fail to generalise to different data (o.o.d. test sets), but the intended solution does generalise.



III. Shortcuts defined: a taxonomy of decision rules

(4) intended solution

- *Intended features* cause model to succeed with o.o.d. tests

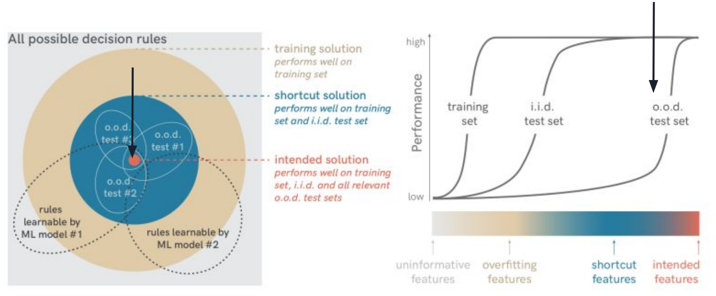
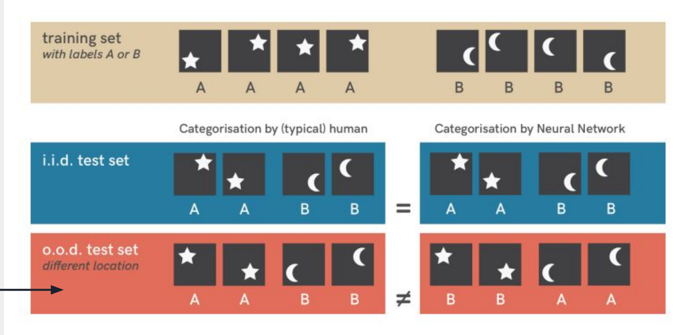

All possible decision rules
training solution
performs well on training set
shortcut solution
performs well on training set and i.i.d. test set
intended solution
performs well on training set, i.i.d. and all relevant o.o.d. test sets
o.o.d. test #1
o.o.d. test #1
o.o.d. test #2
rules learnable by ML model #1
rules learnable by ML model #2
high
low
Performance
training set
i.i.d. test set
o.o.d. test set
uninformative features
overfitting features
shortcut features
intended features

Figure 3. Taxonomy of decision rules. Among the set of all possible rules, only some solve the training data. Among the solutions that solve the training data, only some generalise to an i.i.d. test set. Among those solutions, shortcuts fail to generalise to different data (o.o.d. test sets), but the intended solution does generalise.



IV. Shortcuts: where do they come from?

- *Shortcuts* = decision rules that perform well on i.i.d. tests but fail on o.o.d. tests

- Where do they come from?
    - Data
    - Feature combination

- What do we look for?

IV. Shortcuts: where do they come from?

(1) Dataset: shortcut opportunities

- What makes a cow a cow?

- Shortcut opportunity = systematic relationship between object and *background* or *context*
- *Dataset biases* = shortcut opportunities resulting from natural relationships
  - Not all are obvious to humans

- Can’t we just add more data?
  - Shortcut opportunities do not disappear
  - Data alone rarely constrains a model sufficiently

IV. Shortcuts: where do they come from?

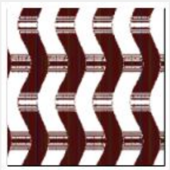
(2) Decision rule: shortcuts from discriminative learning

- What makes a cat a cat?

- *Discriminative learning* = picking a feature that is sufficient to reliably discriminate
- Importance of *feature combination* = definition of an object depends on combination of information from different sources or attributes that influence a decision rule

- Can’t we just shift the data?
  - Excessive invariance can be harmful

IV. Shortcuts: where do they come from?



(3) Generalization: how shortcuts can be revealed

- What makes a guitar a guitar?
- O.o.d. helpful in identifying shortcuts
- Generalization failures are not failures in learning or ability to generalize → failure to generalize in *intended* direction

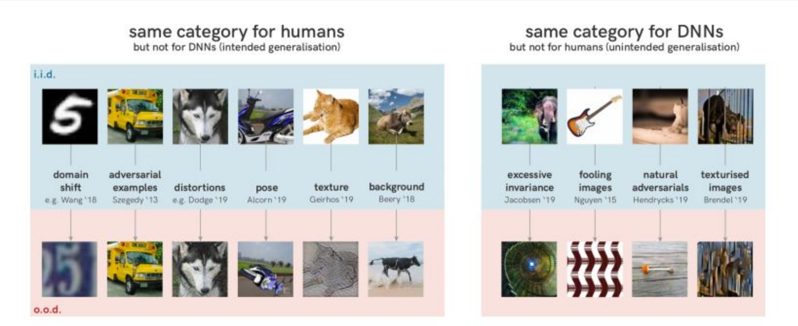


Figure 4. Both human and machine vision generalise, but they generalise very differently. Left: image pairs that belong to the same category for humans, but not for DNNs. Right: images pairs assigned to the same category by a variety of DNNs, but not by humans.

V. Shortcut learning across deep learning

General problems appears across deep learning. Most prominent:

- Computer vision
  - Domain transfer = transferring model performance across datasets
  - Adversarial examples
- Natural language processing
  - BERT and superficial cues words
- Agent based (Reinforcement) learning
  - Reward hacking
  - Generalization gap between simulated and real world
- Fairness and algorithmic decision-making
  - Bias amplification
  - Disparity amplification

VI. Diagnosing and understanding shortcut learning

(1) Interpreting results carefully

- Need to test underlying abilities (not possible with i.i.d.)

- *Anthropomorphism* = human like characteristics to non-humans
- Expand on Morgan’s Canon for machine learning: “never attribute to high-level abilities that which can be adequately explained by shortcut learning”

- Testing (surprisingly) strong baselines

VI. Diagnosing and understanding shortcut learning

(2) Detecting shortcuts: towards o.o.d. generalization tests

- Making o.o.d. generalization tests a standard practice: i.i.d. tests are “the big lie”
- Designing food o.o.d. tests
  - (1) Clear distribution shift
  - (2) Well-defined intended solution
  - (3) Test where current models struggle

- Encouraging examples
  - ImageNet-A: set of wrongly classified images
  - ImageNet-C: images with 15 corruptions (like blur and noise)
  - Shift-MNIST: includes added correlations that shortcuts should pick up

VI. Diagnosing and understanding shortcut learning

(3) Shortcuts: why are they learned?

- Models may use the “Principle of Least Effort”

- Understanding if a solution is easy to learn
  - Structure (architecture) - convolutions make using location harder
  - Experience (training data) - adding data does not get rid of dataset biases
  - Goal (loss function) - Cross entropy encourages use of simple predictors
  - Learning (optimization) - small learning rate can lead to simple patterns, large to complex patterns with memorization

VII. Beyond shortcut learning

O.o.d. generalization lacking across machine learning. Shortcut learning is connected to other areas of research.

- Domain specific knowledge - use data augmentation or rotation invariance
- Adversarial examples - attacks
- Domain adaptation, generalization, randomization - use multiple distributions in training
- Generative modeling - model every variation by generating training data

Conclusion

Just to recap, this paper

- Identifies pattern of shortcuts in BNN and ANN
- Argues need to interpret results carefully
- Pushes for move of testing benchmark toward o.o.d.

VIII. What other research is being done?

Detecting Spurious Correlations With Sanity Tests for Artificial Intelligence Guided Radiology Systems (2021)

- Extend guidelines to identify and address shortcuts with sanity tests:
  - (1) train and test with targets absent and present
  - (2) train and test with background patches or noise
  - (3) train on different regions of interest

Can contrastive learning avoid shortcut solutions? (2021)

- Constrastive learning = uses discrimination that follows shortcuts
- Propose implicit feature modification that encourages using multiple input features (feature combination)

AI for radiographic COVID-19 detection selects shortcuts over signal (2021)

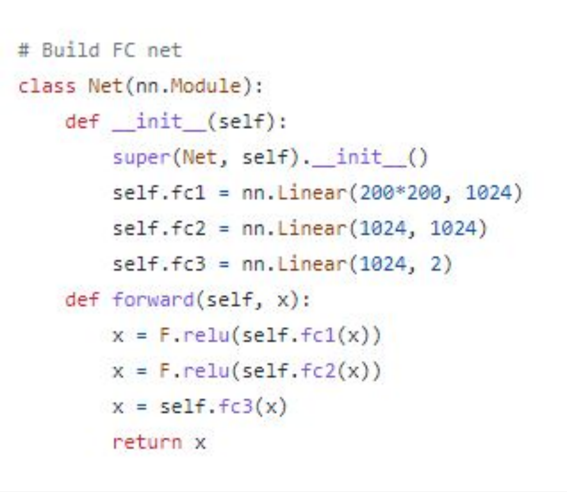
- COVID-19 chest scans are ideal for shortcuts
- Suggest using better data and better numerical quantification of clinical labels

Thank you

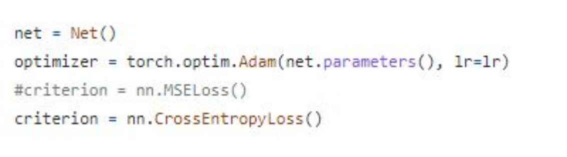
Paper Fully Connected Neural Network implementation (toy example)

- Three-layer ReLU MLP with 1024 units in each layer, two output units corresponding to the two target classes
- Reaches 100% accuracy at training time and chance-level accuracy at test time (51.0%)
- Code available at https://github.com/rgeirhos/shortcut-perspective

- Two easily distinguishable shapes (star and moon) placed on a 200×200 2D canvas
- The training set is 4000 images, where 2000 are star and 2000 moon
- Location of star and moon are controlled in training, but randomized in testing



```
# Build FC net
class Net(nn.Module):
    def __init__(self):
        super(Net, self).__init__()
        self.fc1 = nn.Linear(200*200, 1024)
        self.fc2 = nn.Linear(1024, 1024)
        self.fc3 = nn.Linear(1024, 2)
    def forward(self, x):
        x = F.relu(self.fc1(x))
        x = F.relu(self.fc2(x))
        x = self.fc3(x)
        return x
```



```
net = Net()
optimizer = torch.optim.Adam(net.parameters(), lr=lr)
#criterion = nn.MSELoss()
criterion = nn.CrossEntropyLoss()
```