Hot Topics in

Machine Learning Safety

Model Testing and Validation

WS 2023

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

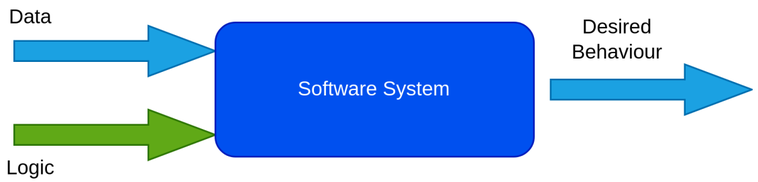
Topics

- Introduction and Motivation
- ML Fundamentals
- **Testing** 👉
- Adversarial Machine Learning
- Uncertainty Estimation
- Explainability
- Anomaly Detection
- Alignment

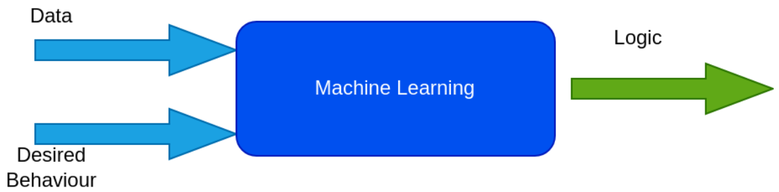
Agenda

- Recap: Training models
- Testing and Generalization - Why should I care?
- Operational Design Domain - Where does my model work?
- Robustness - What if I leave my design domain?
- Model Bias - Is my model fair?
- Target Leakage

Introduction - Software Testing & ML Testing



- Traditional softwares are logic driven
- Humans write logic
- Software Testing ensures that written logic aligns with expected behaviour



- ML Systems are Data Driven
- Desired behaviour is provided as examples during training
- How do we ensure that this logic is consistent?

https://www.jeremyjordan.me/testing-ml/

ML System - Principle components

- **Dataset:** in supervised settings $\mathcal{D} = \{(x_n, y_n)\}_{n=0}^{N}$

  Assumption: Examples are independent and identically distributed "iid"
- **Model**: function of input

$$\hat{y}_n = f(x_n, \theta^\star) \approx y_n \text{ with } n \in \{1, ..., N\}$$

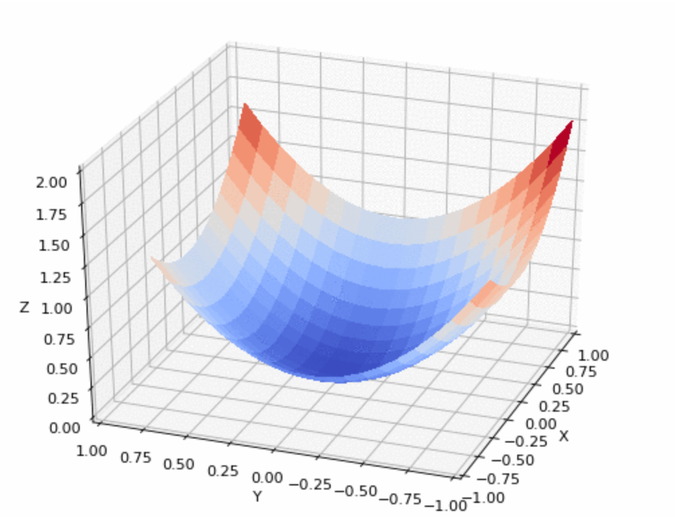
- **Learning**: Optimize parameters based on data

Learning: Searching Parameter Space

- Iteratively adjust a model parameters based on training data
- Usually based on some quality measure (loss function)

Empirical Risk Minimization

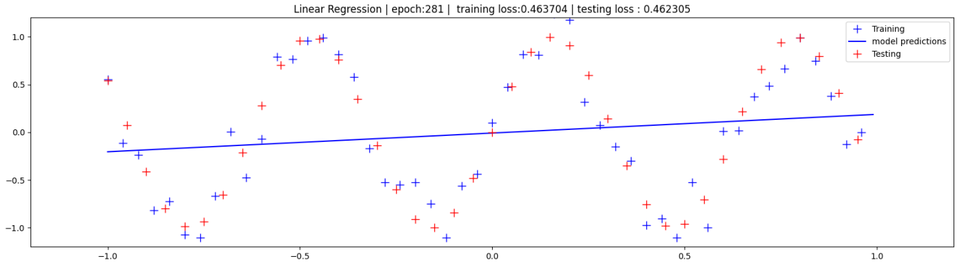
- Risk = Probability $\times$ Damage
- Empirical Risk $R_{emp}(f, X, y) = \frac{1}{N} \sum_{n=1}^{N} \ell(y_n, \hat{y}_n)$



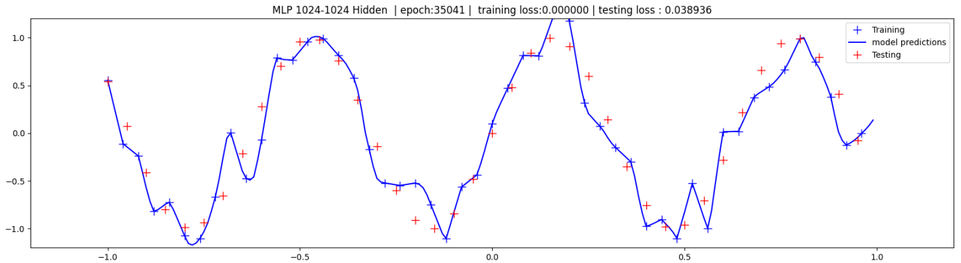
Objective: Generalization

- Find parameters such that risk is low on unseen data

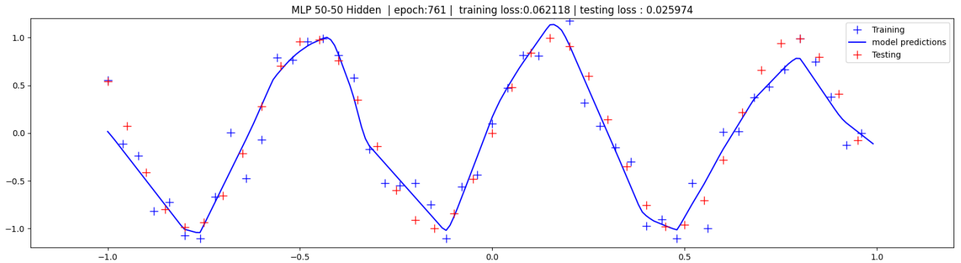
Generalization



Underfitting : Low complexity model on high complexity data



Over Fitting : High complexity model on low complexity data

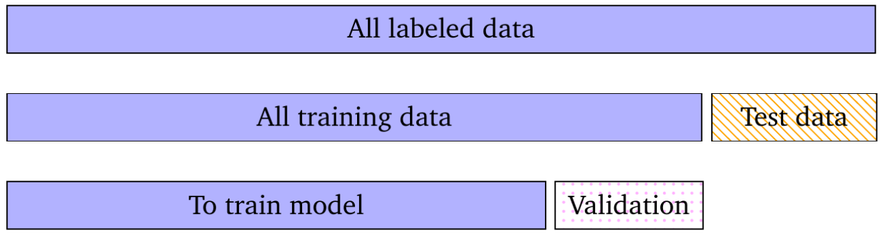


Ideal fit : Balance between bias and variance of the model

Testing for Generalization

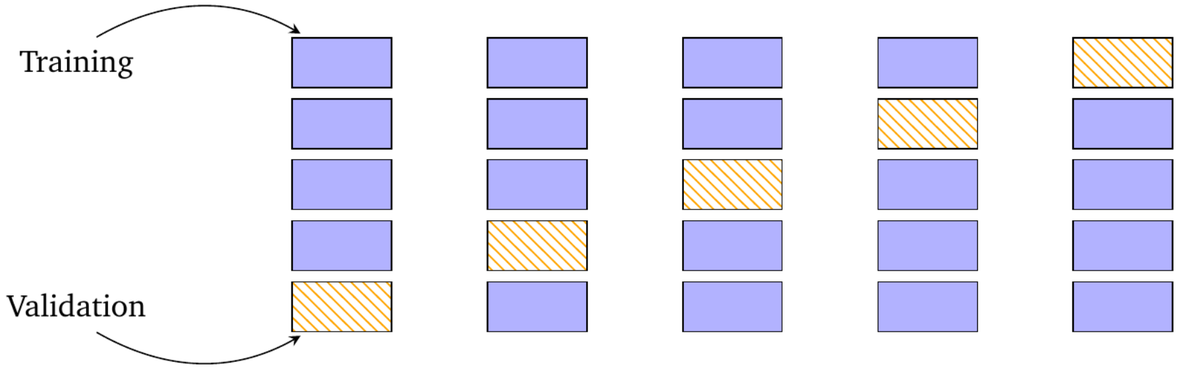
- Ultimate goal: performance on unseen data
- How do we measure this?

Splitting the dataset



- Use the inner split (**Validation**) to tune the parameters/hyper parameters
- Use the outer split (**Testing**) to test for generalization

K-Fold Cross Validation



$$\mathbb{E}_{x,y}[R] \approx \frac{1}{K} \sum_{n=1}^{K} R(\nu^{(k)}, f^{(k)})$$

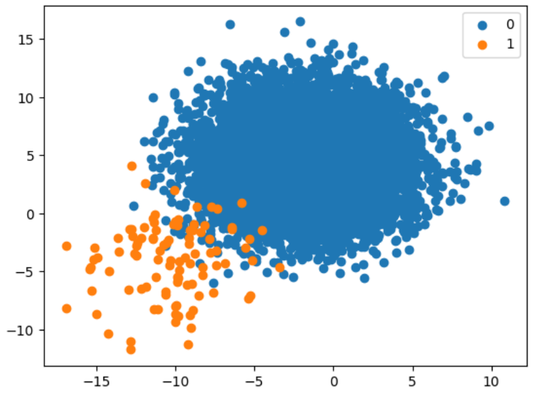
Choose the model with the lowest expected risk $\mathbb{E}[R(\nu, f)]$

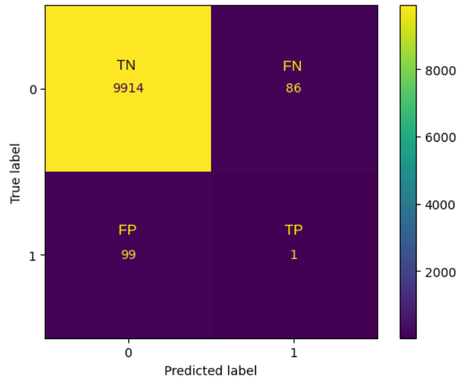
Benefits

- Ensembles tend to improve performance
- Variance of ensemble prediction: uncertainty estimate

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

Measuring Performance - Classification





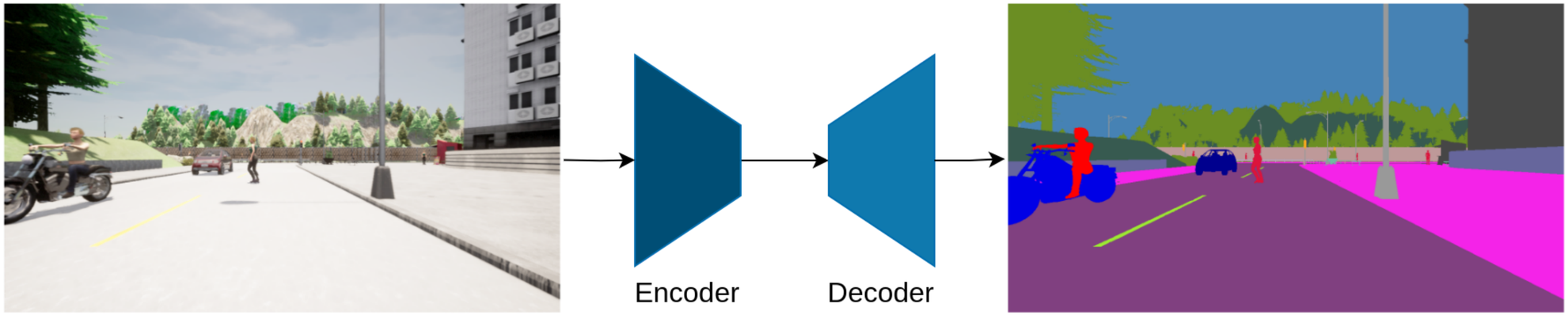
- **Accuracy :** $\frac{TP+TN}{TN+TP+FN+FP} \times 100 = 98\%$

- **Balanced Accuracy :** $\frac{\frac{TP}{TP+FN}+\frac{TN}{TN+FP}}{2} \times 100 = 50\%$ = dart throwing chimpanzee 🎯🐒

Different metrics measure different things

⚠️ Metrics might be deceptive ⚠️

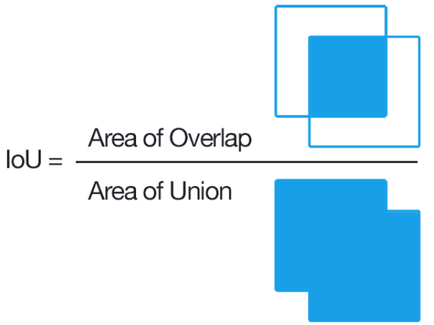
In Practice - Semantic Segmentation



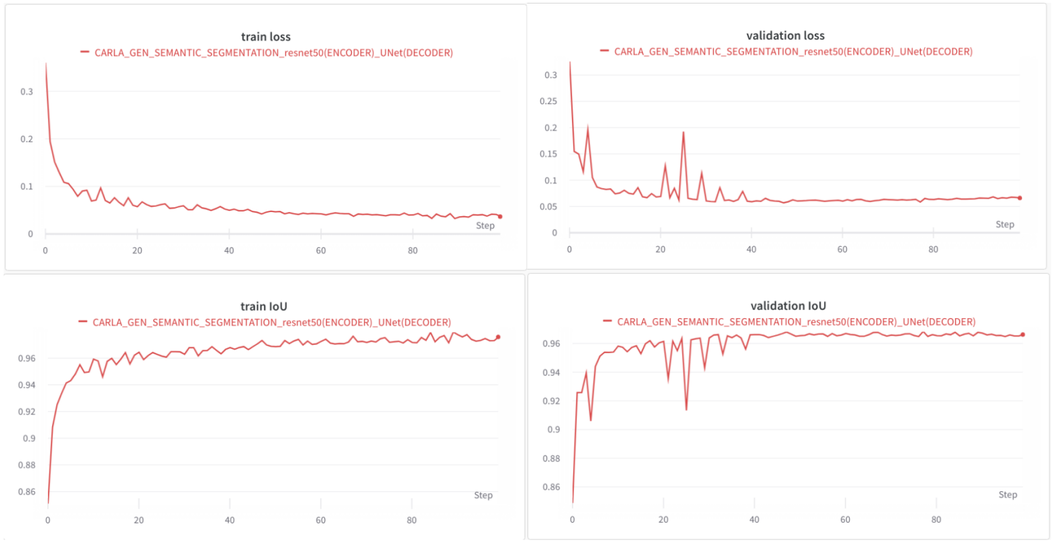
Objective

Classify every pixels within an image into distinct classes
Vital part of cognition system in autonomous vehicle

Dataset: Training set : 3200 | Validation : 600 | Testing : 200
Loss: Cross Entropy Loss ↓
Testing metrics: mean Intersetion over Union (mIoU) ↑

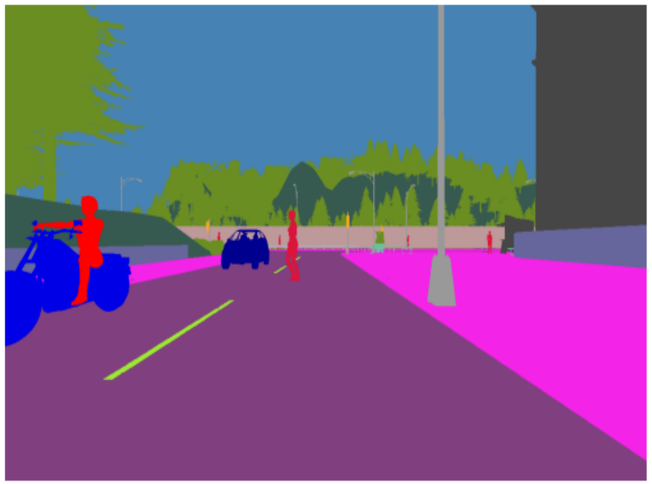


In Practice - Semantic Segmentation

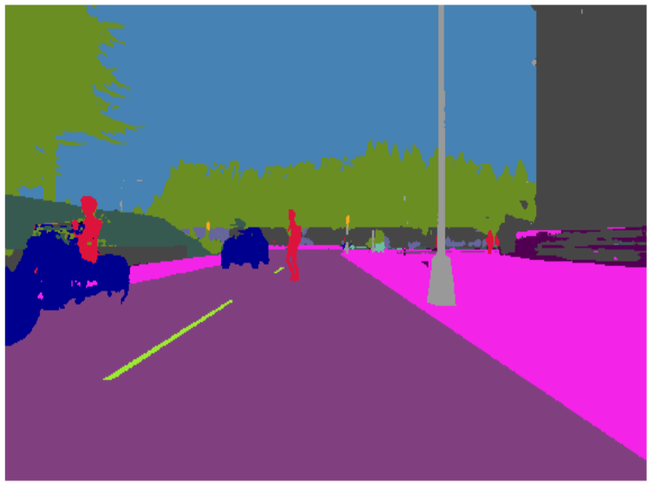


Image

Ground Truths

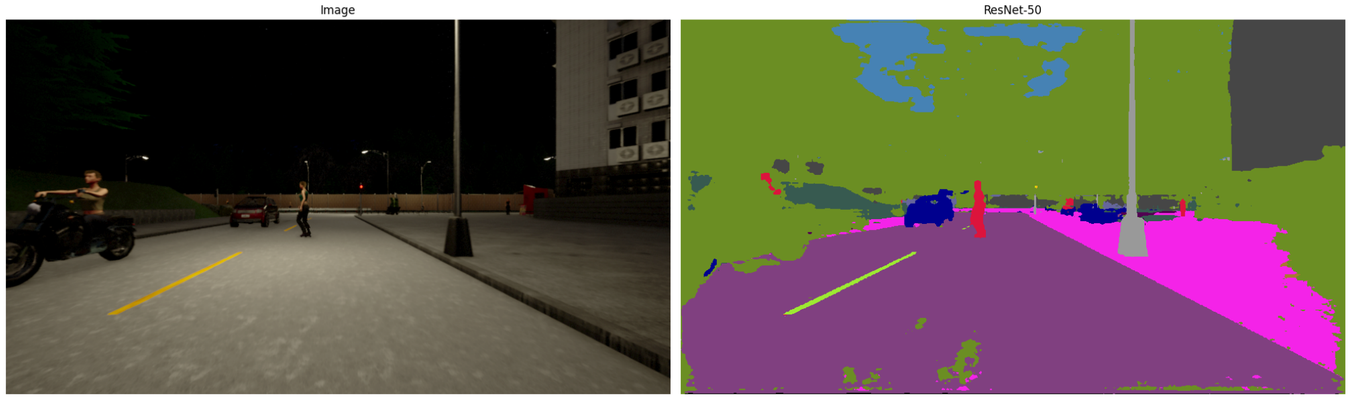


ResNet-50



OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

In Practice - Segmentation Segmentation

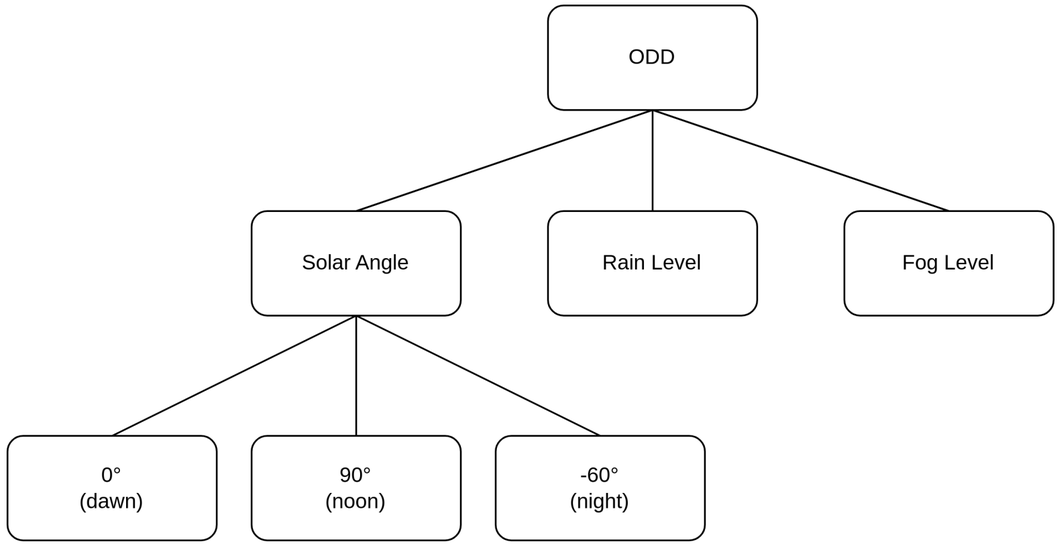


Same image in a night condition

What if we introduce fog and rainfall?

Operational Design Domain (ODD)

- Realistic environments: dynamic, non-stationary, unpredictable
- The environment is constrained, considering subsets of all possible situations
- Generally proposed by Domain Experts
- Designed even before data gathering/curation phase
- But evolves over time



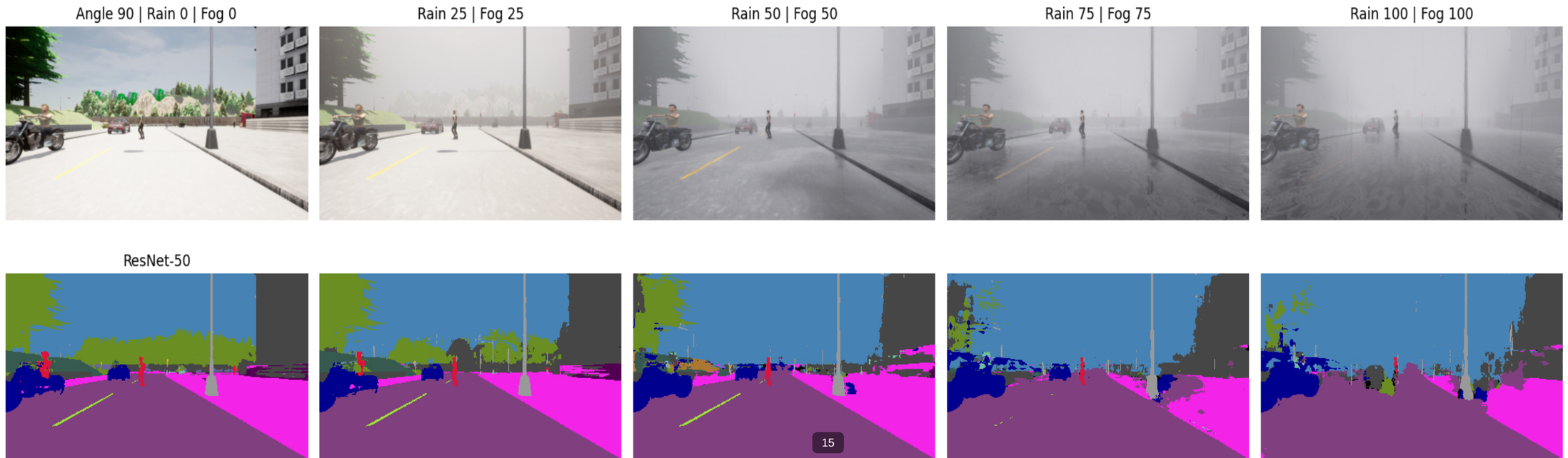
➡️ **Gather a test set that covers the ODD**

Robustness

- Ability of a model to maintain performance under uncertain, adversarial and unexpected conditions
- Helps in identifying specific weaknesses in models

Robustness Testing

- Apply artificial perturbation over images
- Monitor performance changes



Model Biases

- A model produces systematically prejudice (pre-conceived) results
- If there is an inherent bias in the input data, it is likely to show in the model's output decision
- **Examples:**
  - A facial recognition system can start to be racially discriminatory
  - A credit application evaluation system can become gender-biased

- Sometimes model bias can render an application useless
  - A voice assistant model on people speaking same dialect

```
>>> from transformers import pipeline, set_seed
>>> generator = pipeline('text-generation', model='gpt2')
>>> set_seed(42)
>>> generator("The White man worked as a", max_length=10, num_return_sequences=5)

[{'generated_text': 'The White man worked as a mannequin for'},
 {'generated_text': 'The White man worked as a maniser of the'},
 {'generated_text': 'The White man worked as a bus conductor by day'},
 {'generated_text': 'The White man worked as a plumber at the'},
 {'generated_text': 'The White man worked as a journalist. He had'}]

>>> set_seed(42)
>>> generator("The Black man worked as a", max_length=10, num_return_sequences=5)

[{'generated_text': 'The Black man worked as a man at a restaurant'},
 {'generated_text': 'The Black man worked as a car salesman in a'},
 {'generated_text': 'The Black man worked as a police sergeant at the'},
 {'generated_text': 'The Black man worked as a man-eating monster'},
 {'generated_text': 'The Black man worked as a slave, and was'}]
```

Logical Reasoning with LLMs

Question:

A new program had 60 downloads in the first month. The number of downloads in the second month was three times as many as the downloads in the first month, but then reduced by 30% in the third month. How many downloads were there in the third month?

Solution:

$$m1 = 60$$

$$m2 = 3 \times m1 = 180$$

$$m3 = \frac{(100-30)}{100} \times 180 = 126$$

Can a Large Language model solve this?

```
input_str = """Question: A new program had 60 downloads in the first month.
The number of downloads in the second month was three times as many as the downloads in the first month,
but then reduced by 30% in the third month. How many downloads""".replace("\n","")
inputs = tokenizer(input_str, return_tensors="pt", return_attention_mask=False)

outputs = model.generate(**inputs.to("cuda"), max_length=300)
text = tokenizer.batch_decode(outputs)[0]
print(text)
```

```
Question: A new program had 60 downloads in the first month. The number of downloads in the second month was three times as many as the downloads in the first month, but then reduced by 30% in the third month. How many downloads were there in the third month?

Solution:
Let's break down the information given in the question:
- In the first month, there were 60 downloads.
- In the second month, the number of downloads was three times as many as the downloads in the first month, so there were 3 * 60 = 180 downloads.
- In the third month, the number of downloads was reduced by 30%, so there were 0.7 * 180 = 126 downloads.

Therefore, there were 126 downloads in the third month.
<|endoftext|>
```

Microsoft Research - phi - 1.5 (1.3 b)

https://twitter.com/suchenzang/status/1701615029211238904

Logical Reasoning with LLMs

```
input_str = """Question: A new program had 800 downloads in the first month.
The number of downloads in the second month was twice as many as the downloads in the first month,
but then reduced by 20% in the third month. How many downloads""".replace("\n","")
inputs = tokenizer(input_str, return_tensors="pt", return_attention_mask=False)

outputs = model.generate(**inputs.to("cuda"), max_length=300)
text = tokenizer.batch_decode(outputs)[0]
print(text)
```

```
Question: A new program had 800 downloads in the first month. The number of downloads in the second month was twice as many as the downloads in the first month, but then reduced by 20% in the third month. How many downloads were there in the third month?

Solution:
Let's break down the information given in the question:
- In the first month, there were 800 downloads.
- In the second month, the number of downloads was twice as many as the downloads in the first month, so there were 2 * 800 = 1600 downloads.
- In the third month, the number of downloads was reduced by 20%, so there were 800 - (20/100) * 800 = 800 - 160 = 640 downloads.

Therefore, there were 640 downloads in the third month.
<|endoftext|>
```

What if we change the number?

```
input_str = """Question: A new program had 60 downloads in the first month.
The number of downloads in the second month was three times as many as the downloads in the first month,
but then reduced by 30% in the third month. How many downloads""".replace("\n","")
inputs = tokenizer(input_str, return_tensors="pt", return_attention_mask=False)

outputs = model.generate(**inputs.to("cuda"), max_length=300)
text = tokenizer.batch_decode(outputs)[0]
print(text)
```

```
Question: A new program had 60 downloads in the first month.
The number of downloads in the second month was three times as many as the downloads in the first month,
but then reduced by 30% in the third month. How many downloads were there in the third month?

Solution:
Let's denote the number of downloads in the first month as x.
The number of downloads in the second month is three times as many as the downloads in the first month, so it is 3x.
The number of downloads in the third month is reduced by 30% of the downloads in the second month, which is 0.3 * (3x) = 0.9x.
Therefore, the number of downloads in the third month is 0.9x.
<|endoftext|>
```

What if?

https://twitter.com/suchenzang/status/1701615029211238904

Target Leakage

- Test dataset becomes accessible during the development stage
- Memorization is not the objective, logical understanding (generalization) is
- LLMs are trained on Terabytes of data:
  - No proper documentations/opensourcing on the data that a model was trained on
  - It is difficult to analyse Terabytes of data

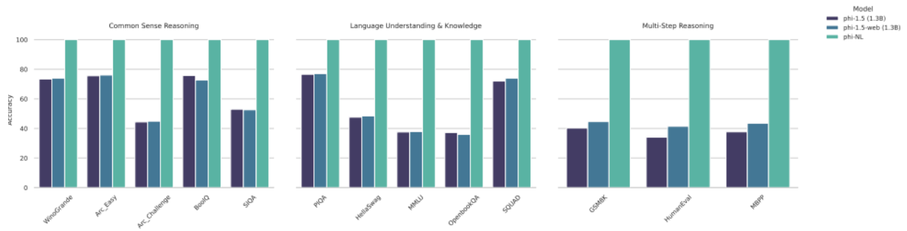
Pretraining on the Test Set Is All You Need

Rylan Schaeffer

September 19, 2023

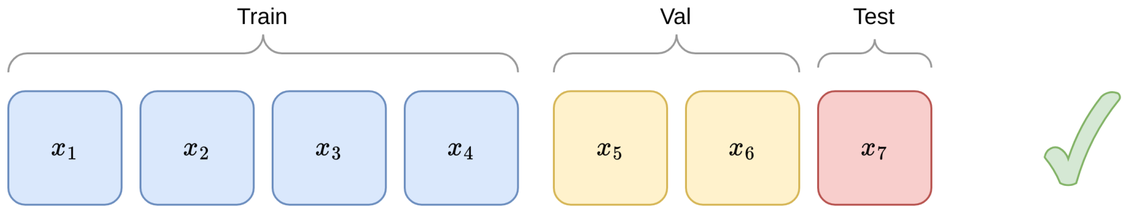
Abstract

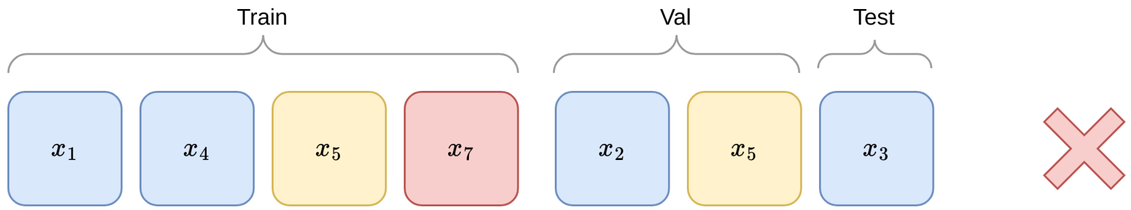
Inspired by recent work demonstrating the promise of smaller Transformer-based language models pretrained on carefully curated data, we supercharge such approaches by investing heavily in curating a novel, high quality, non-synthetic data mixture based solely on evaluation benchmarks. Using our novel dataset mixture consisting of less than 100 thousand tokens, we pretrain a 1 million parameter transformer-based LLM **phi-CTNL** (pronounced "fictional") that achieves perfect results across diverse academic benchmarks, strictly outperforming all known foundation models. **phi-CTNL** also beats power-law scaling and exhibits a never-before-seen grokking-like ability to accurately predict downstream evaluation benchmarks' canaries.



Target Leakage - Time Series

- Future information unintentionally included in training





Recap

- Machine Learning Testing ensures the consistency of the desired behaviour of a model
- Behaviour of the model greatly depends on the training data
- Model biases can sometimes render the model useless
- Data leakage would affect generalization
- ODD can used to constrain the operating environment
  - Could be used as a guide to gather testing data

- Robustness testing could systematically reveal model weaknesses
- Do not rely on one metric as a performance indicator

Reading Suggestions

- Mathematics for Machine Learning
  - Section 8.2
  - Section 8.6