Query languages for Neural Networks

A Black-box & White-box logic

November 22, 2024

The current state of Neural Network management

"Let me quickly write that in Python"

- User creates their own model and trains it
- NN lives as a file on machine
- Not easily searchable in structured manner

AutoML

- Trains and selects best model for user
- Done separately for each model
- ML model usually opaque to user

Model Registries

- User uploads trained model
- Simple searching based on metadata

Queries for different management systems

"Give me all models with at least 70000 parameters"

"Let me quickly write that in Python"

Queries for different management systems

"Give me all models with at least 70000 parameters"

AutoML

Queries for different management systems

"Give me all models with at least 70000 parameters"

Model Registries

Queries for different management systems

"Give me the closest counterfactual for this input"

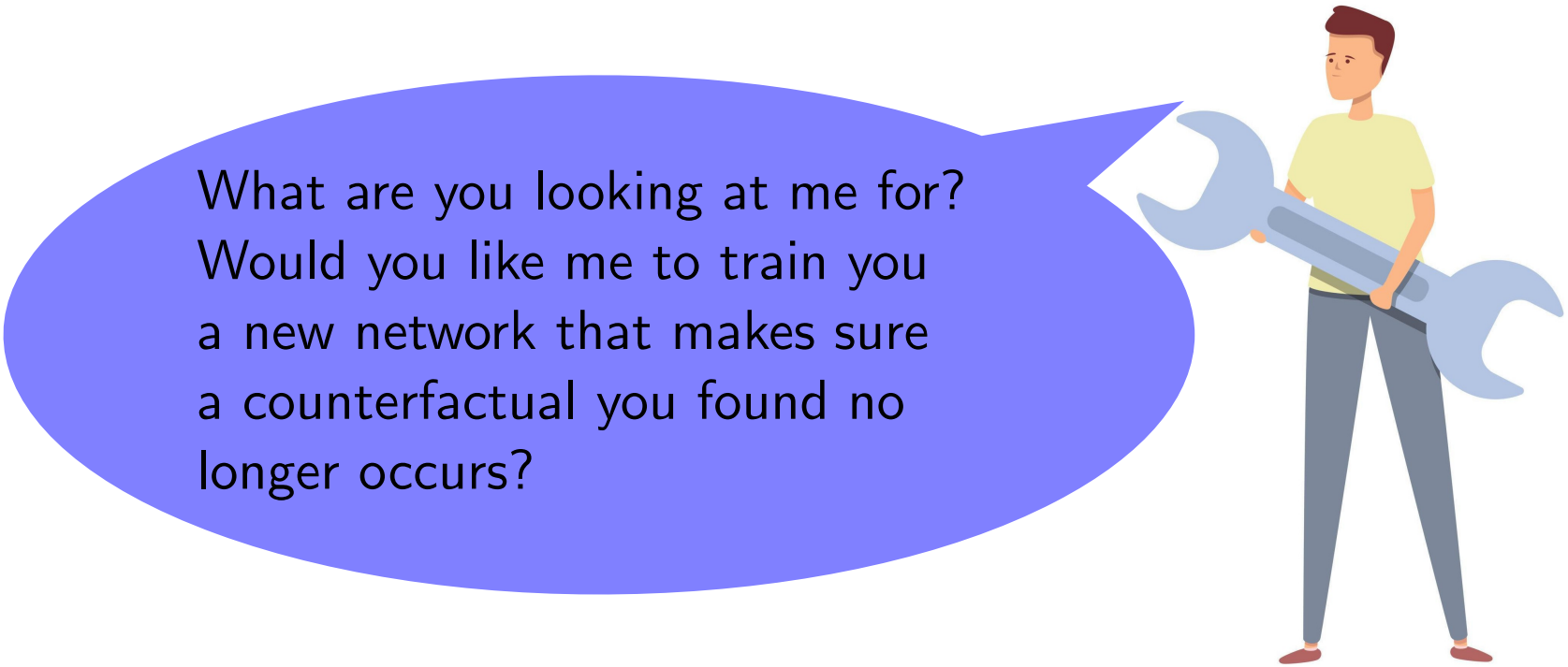
"Let me quickly write that in Python"

Now this sounds like a fun little project in which you spend the next few weeks learning how to implement or use the built-in backpropagation to find a solution to this. Or we can slowly explore more and more of the input space until we find a counterfactual and work from there! Or....

Queries for different management systems

"Give me the closest counterfactual for this input"

AutoML



Queries for different management systems

"Give me the closest counterfactual for this input"

Model Registries

Would you maybe like to look for a model that you tagged with a tag called 'counterfactual'?

Queries for different management systems

"Give me all the networks containing a convolutional layer"

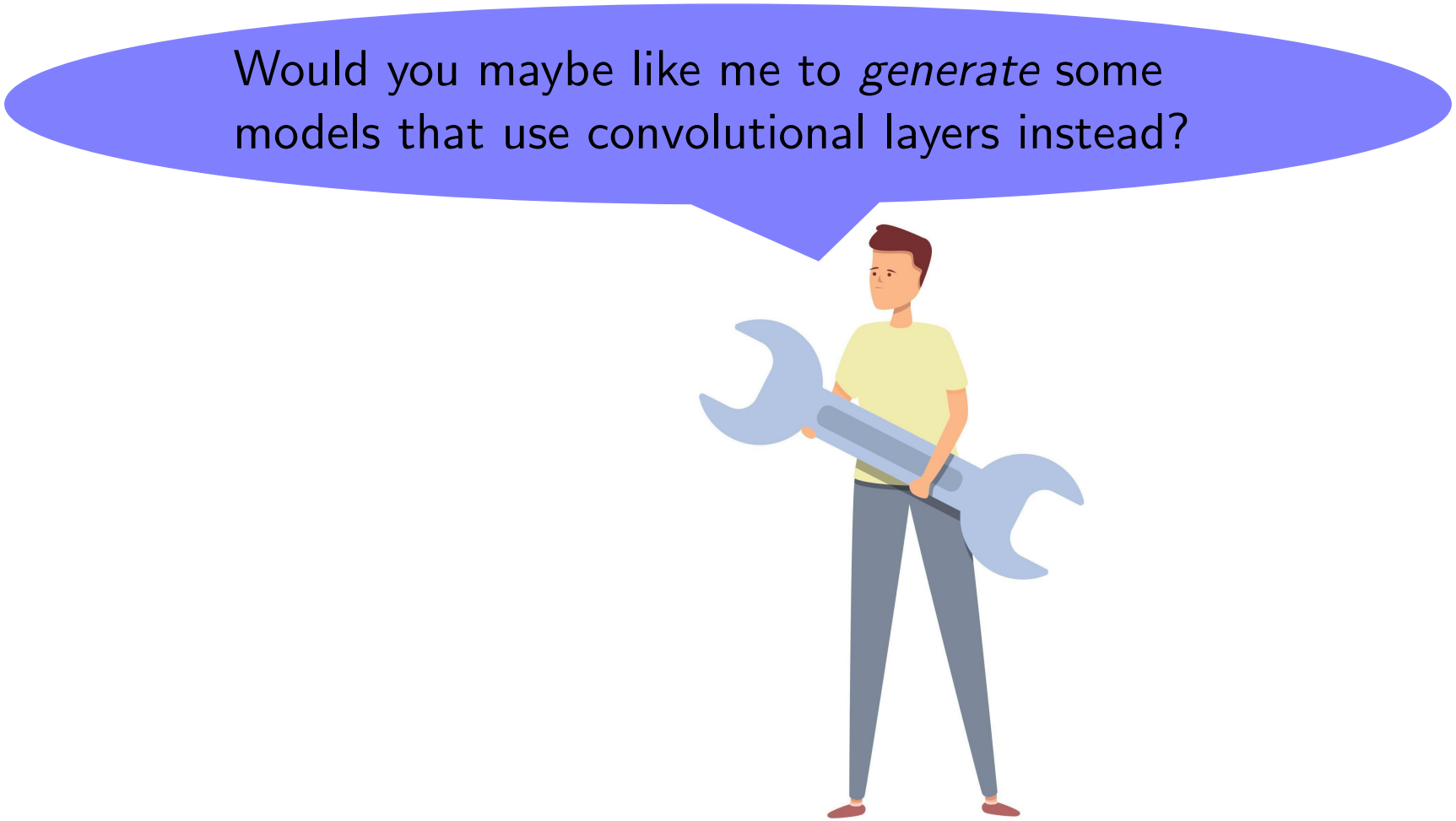
"Let me quickly write that in Python"

Hmmmm, maybe we should search through all the files and try to see if any of them contain the string of the class of the convolution layer... Or maybe we should try to load all the models from these files and see if we can identify a convolution layer using the specific classes of our library?

Queries for different management systems

"Give me all the networks containing a convolutional layer"

AutoML



Queries for different management systems

"Give me all the networks containing a convolutional layer"

Model Registries

Convolutional layers are mostly used in image data, right? Would you like to me to look for all models that use images as their input?

Queries for different management systems

Conclusion

Most queries cannot be done by these systems or require lots of implementation work

- Similar to state of databases before relational databases
- Why not use a declarative query language?

Two logics

FO(**R**,F) (Black-box Logic)

FOL over Reals + model function **(constraint query language)**

- + quantifying over the reals
- − model is given as black box
- − no guarantees on runtime

FO(SUM) (White-box logic)

FOL over k-relations + SUM aggregate on model structure **(SQL)**

- − cannot quantify over the reals
- + can inspect internals of model
- + guaranteed to finish

Two logics

Expressiveness

What different kinds of queries can these logics answer?

Common ground

- Evaluated on single model

Strenghts of the languages

Black-box Logic: Questions about shape & form of model function

White-box Logic: Questions about internal structure of model

Queries revisited

"Does *this model* have at least 70000 parameters"

Black-box Logic: Cannot be expressed

White-box Logic: $((\sum_{a,b:w(a,b)\neq\perp} 1) + \sum_{a:b(a)\neq\perp} 1) \geq 70000$

"Give me the closest counterfactual for this input"

Black-box Logic: Assuming we can express distance metric d

$$\exists x_1' \ldots x_n' (|F(x_1', \ldots, x_n') - F(x_1, \ldots, x_n)| > \delta$$
$$\wedge \neg(\exists x_1'' \ldots x_n'' (|F(x_1', \ldots, x_n') - F(x_1, \ldots, x_n)| > \delta$$
$$\wedge d(x_1, \ldots, x_n, x_1'', \ldots, x_n'') < d(x_1, \ldots, x_n, x_1', \ldots, x_n'))))$$

White-box Logic: Cannot be expressed

Queries revisited

"Does *this network* contain a convolutional layer"

Black-box Logic: Cannot be expressed

White-box Logic: Can be expressed for any kernel size assuming a straightforward encoding of CNN layers into FNN layers

Other queries

"Derivative at given point"

Black-box Logic: Can be expressed for any point

White-box Logic: Can be expressed for ReLU-FNN

"Integral for given range"

Black-box Logic: Cannot be expressed

White-box Logic: Can be expressed for ReLU-FNN

"SHAP score for given input"

Black-box Logic: Cannot be expressed

White-box Logic: Can be expressed for ReLU-FNN

BBL vs FO(SUM)

Representation independent queries

In general incomparable $\rightarrow$ find interesting query class to compare

Representation independent queries

- Binary queries
- Query can only depend on function of model, not structure
- Models with same function, but different structure have same result

Examples

70000 parameters : No, representation dependent

Closest counterfactual : **Yes, representation independent**

Convolutional layer : No, representation dependent

Derivative : **Yes, representation independent**

Integral : **Yes, representation independent**

SHAP score : **Yes, representation independent**

BBL vs FO(SUM)

In general

- FO(SUM) $\not\subseteq$ BBL: FO(SUM) can express representation dependent queries, BBL cannot
- **BBL $\subset$ FO(SUM) (Main result)**:
  - **for models using only ReLU? At least for FO+LIN BBL queries**
  - for models using well-behaved activation functions? Likely, with same/similar restrictions
  - for models any activation function? Who knows

For representation independent queries

- FO(SUM) $\not\subseteq$ BBL: FO(SUM) can express the even peaks and the integration query, BBL cannot
- BBL $\subset$ FO(SUM): from general case (BBL expresses only representation independent queries)

BBL vs FO(SUM)

Consequences of BBL $\subset$ FO(SUM) proof

- Identification of a class of queries that can be **efficiently evaluated**
- **Constructive proof** allowing generation of FO(SUM) query
- Allows for evaluation in existing SQL databases to make use of **existing optimizations techniques**

Conclusions

- Showed **need for a declarative query language**
- Introduced **two possible query languages**
- **Compared expressive power** of query languages
- Determined a **subclass** of BBL expressions that can be **efficiently evaluated**

Questions?

Queries revisited

"Does *this network* contain a convolutional layer"

Black-box Logic: Cannot be expressed

White-box Logic: For all layers i ask

$$\forall n(\text{layer}_i(n) \qquad \text{(Check convolution multiplication structure)}$$

$$\implies \exists n_1 \dots n_k((\bigwedge_{i\in\{1,\dots,k\}} E(n_i, n)) \wedge (\bigwedge_{i\in\{1,\dots,k+1\}} (\bigwedge_{j\in\{i+1,\dots,k+1\}} n_i \neq n_j))$$

$$\wedge \forall n'((\bigwedge_{i\in\{1,\dots,k\}} n_1 \neq n') \wedge \text{layer}_{i-1}(n)) \implies \neg E(n', n)))$$

$$\wedge \forall n(\text{layer}_{i-1}(n) \qquad \text{(Check input for } < k+1 \text{ convolutions)}$$

$$\implies \neg\exists n_1 \dots n_{k+1}((\bigwedge_{i\in\{1,\dots,k\}} E(n, n_i))$$

$$\wedge (\bigwedge_{i\in\{1,\dots,k+1\}} (\bigwedge_{j\in\{i+1,\dots,k+1\}} n_i \neq n_j))))$$