Deep Neural Networks and Discrete Choice Models (Part 2)

Shenhao Wang
191029

Outline

1

Recap

(5 min)

2

Generic Methods for ML Interpretability

(15 min)

3

Architectural Design of DNNs with Alternative-Specific Utility Functions

(30 min)

4

Guest Lecture, Hai Wang

(30 min)

Part 1. Recap

Part 2. Generic Framing & Methods for Interpretable ML

Interpretability is important because of

- Trust
- Safety (e.g. AVs detect objects)
- Transparent governance (e.g. evaluate the eligibility for food stamps)
- New knowledge generation (e.g. academia)

Improving ML Interpretability

Before building models

- Exploratory data analysis (dimension reduction, etc.)

Building a new model

- Use an interpretable model (DT, KNN, etc.)
- Sparsity (e.g. linear model with hundreds of features, etc.)

After building a model (Post-hoc interpretation for DNN)

- Case-based methods
- Gradient-based methods
- Local explanation models
- Global explanation models
- Visualize hidden layers

Improving ML Interpretability

Before building models

- Exploratory data analysis (dimension reduction, etc.)

Building a new model

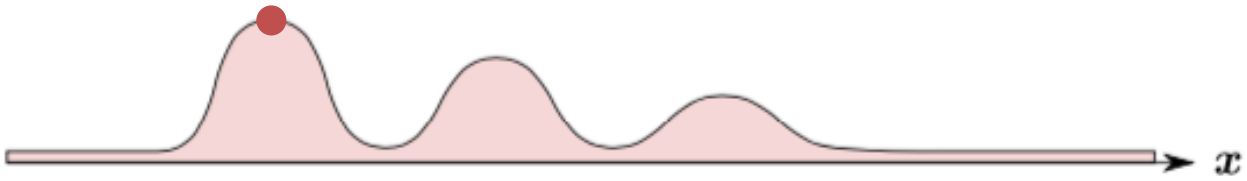
- Use an interpretable model (DT, KNN, DCM)
- Sparsity

After building a model (**Post-hoc interpretation for DNN**)

1. Case-based methods
2. Gradient-based methods
3. Local explanation models
4. Global explanation models
5. Visualize hidden layers

1. Case-based methods: activation maximization (AM)

$$\max_{x} \log P(y = i|x)$$

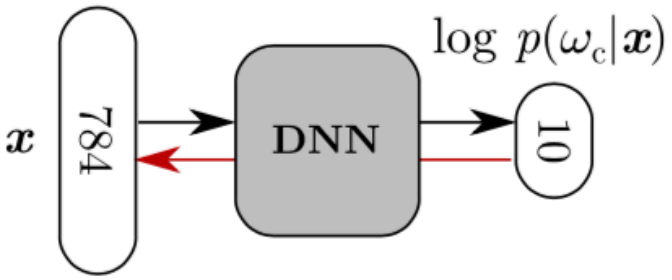


Example

i = auto; AM method can find the representative driver. (prototype)

1. Case-based methods: activation maximization (AM)

$$\max_{x} \log P(y = i|x)$$



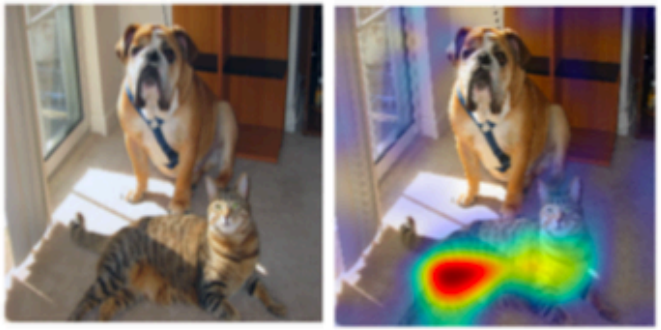
DNN architecture applied to MNIST

Ten prototypes

2. Gradient-based methods (saliency/attribution maps)

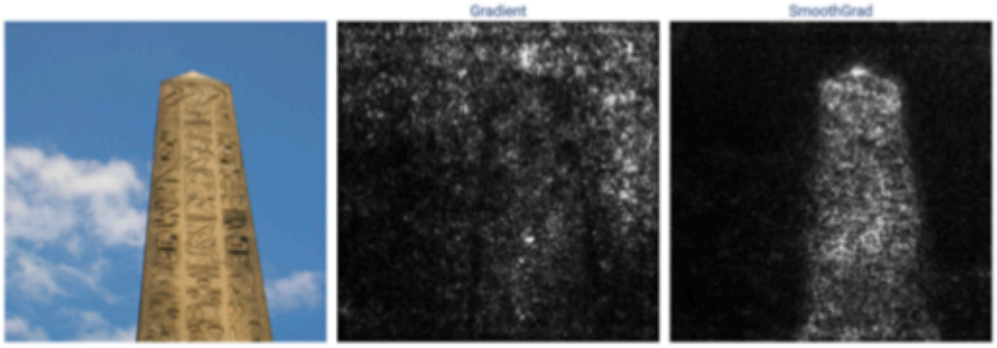
$$\frac{\partial y}{\partial x_{ij}}$$

Grad-CAM [Selvaraju et al. 16]



(a) Original Image (c) Grad-CAM 'Cat'

SmoothGrad [Smilkov et al. 17]



3. Local explanation model (Ribeiro et al., 2016)

Step 1: train a complex model to fit data.

Step 2: train an interpretable model to fit the step1 model around one target.

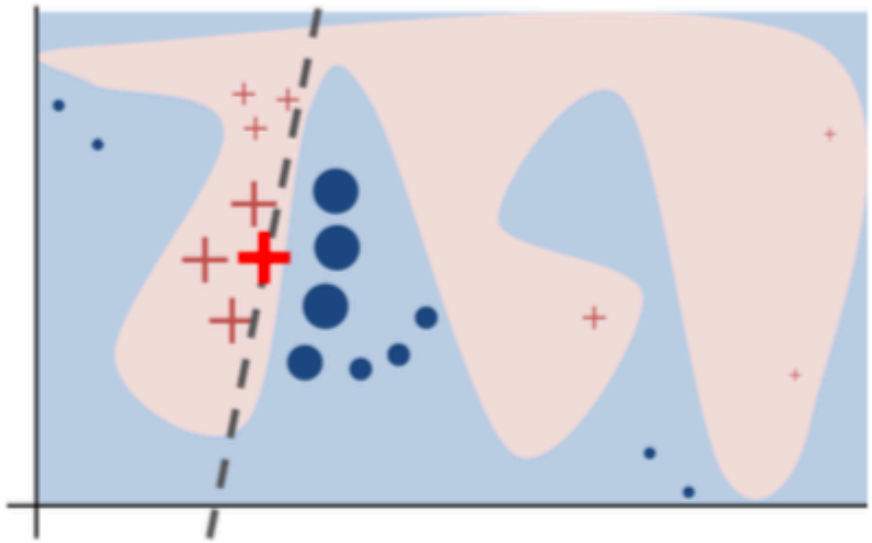
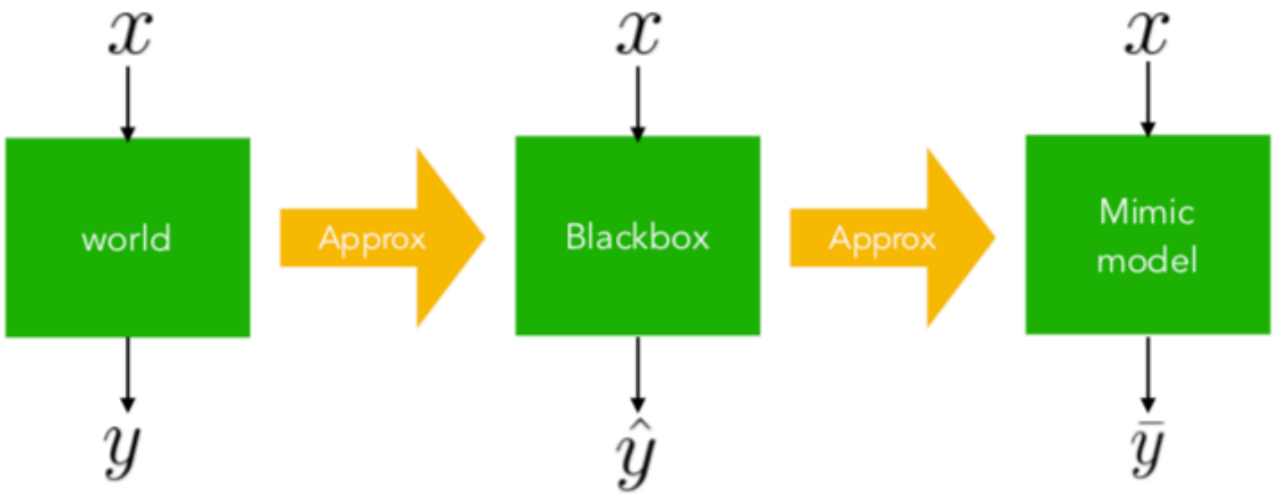


Figure 3: Toy example to present intuition for LIME. The black-box model's complex decision function f (unknown to LIME) is represented by the blue/pink background, which cannot be approximated well by a linear model. The bold red cross is the instance being explained. LIME samples instances, gets predictions using f, and weighs them by the proximity to the instance being explained (represented here by size). The dashed line is the learned explanation that is locally (but not globally) faithful.

4. Global explanation model (Hinton et al., 2015)

Step 1: Train a DNN model to fit the dataset.

Step 2: Train an interpretable model to re-fit the soft labels (choice probability functions) obtained by the step1 DNN model. e.g. use MNL to fit DNN



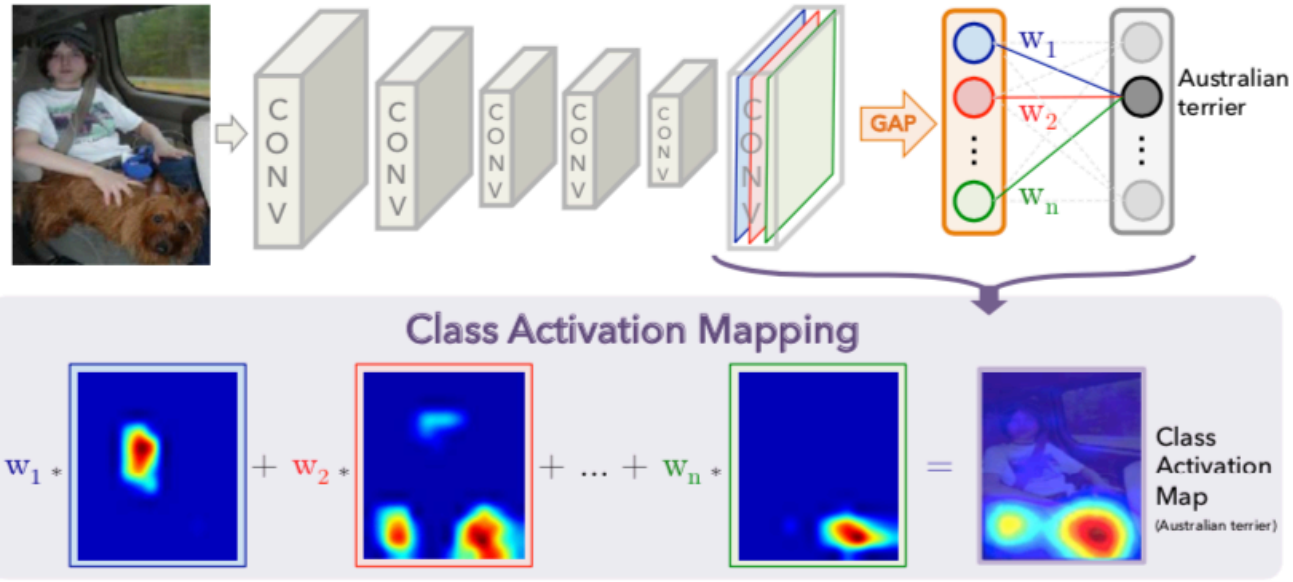
Q: in both local and global explanation models, why not directly use an interpretable model to fit the data, but use a two-step training method?

A: The model trained in the first step captures more information than the data: (Hinton, etc. 2015)

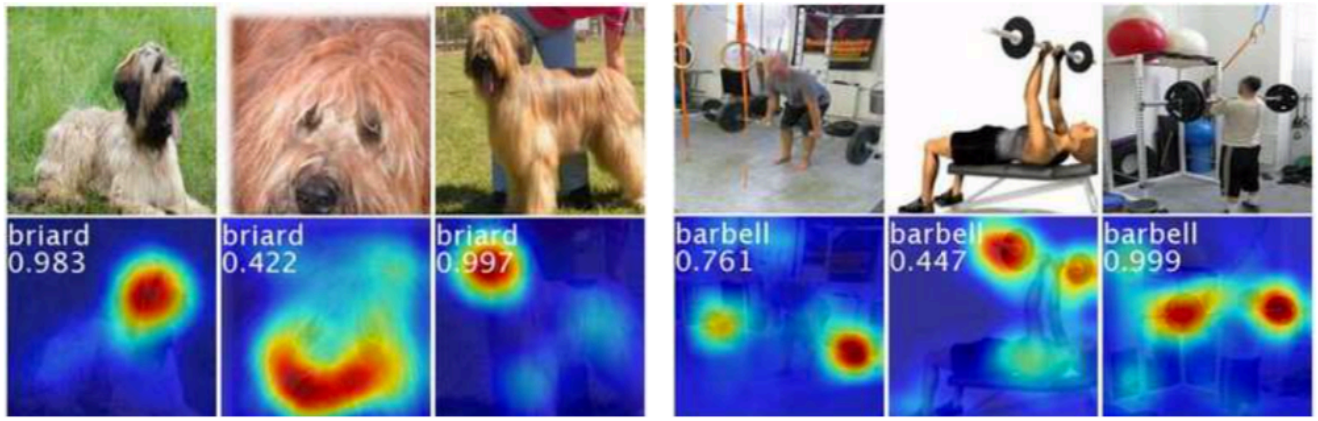
[1, 0, 0, 0] ----- [0.8, 0.1, 0.05, 0.05]

5. Visualize hidden layers (Zhou et al., 2016)

CNN Architectural Design



Visualization for Interpretation



The Mythos of model interpretability (Lipton, 2016)

What is interpretability? Very unclear.

Simulatability (e.g. linear vs. DNN)
Decomposability (e.g. linear model with complicated feature engineering)
Algorithmic transparency (e.g. DNN)
Post-hoc interpretability (e.g. DNN)

e.g. DNNs are post-hoc interpretable, but are not simulatable (in mind).

Future interpretability questions for choice analysis

1. In choice modeling, what are the other pieces of information we can extract from DNN beyond economic information?
   - Case-based methods
   - Gradient-based methods
   - Local explanation models
   - Global explanation models
   - Visualize hidden layers

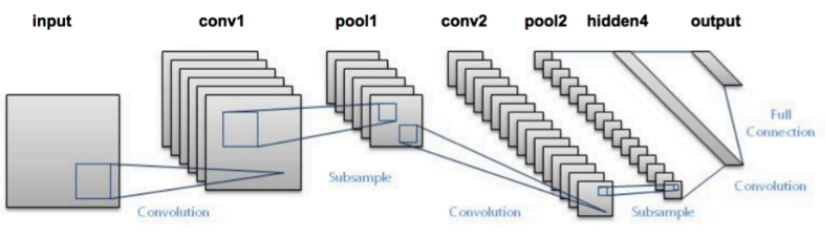
2. Beside post-hoc interpretation, can we design DNN models so that they are ex-ante interpretable?
   - Debate: which way is correct?

Part 3. Architectural Design with Alternative-Specific Utility Functions

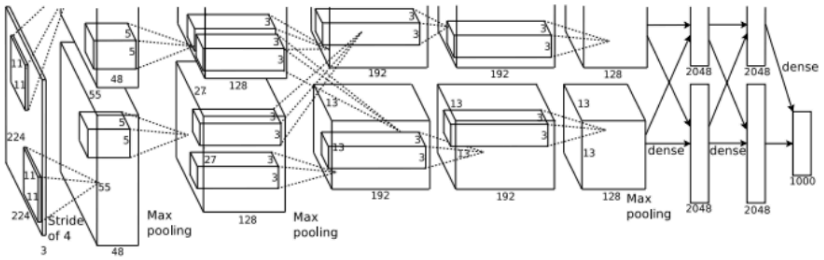
From post-hoc to ex-ante interpretable DNN

Architectural Design of DNNs

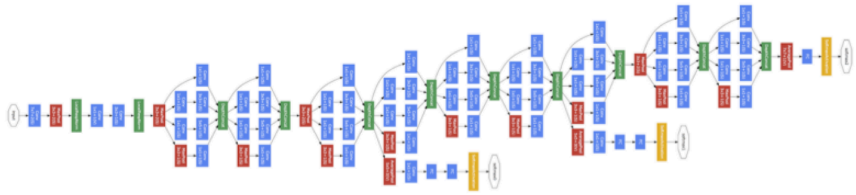
LeNet, 1998



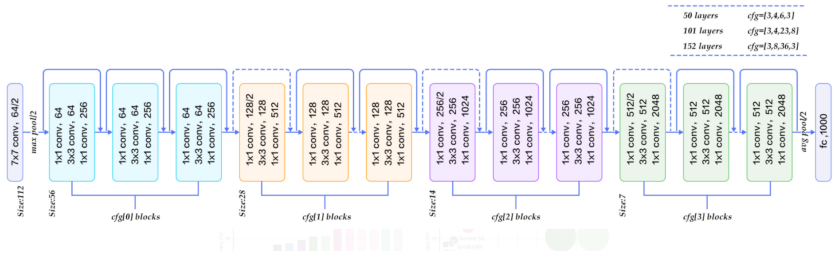
AlexNet, 2012



GoogLeNet, 2014

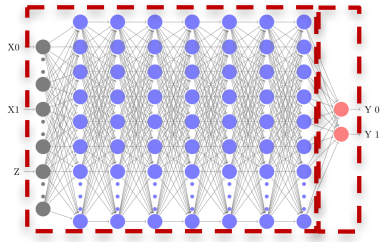


ResNet, 2015



1. Architectural design is critical for DNN performance.
2. However, the DNN is not architecturally interpretable.

Use some utility specification insights to design new DNN architectures for choice analysis



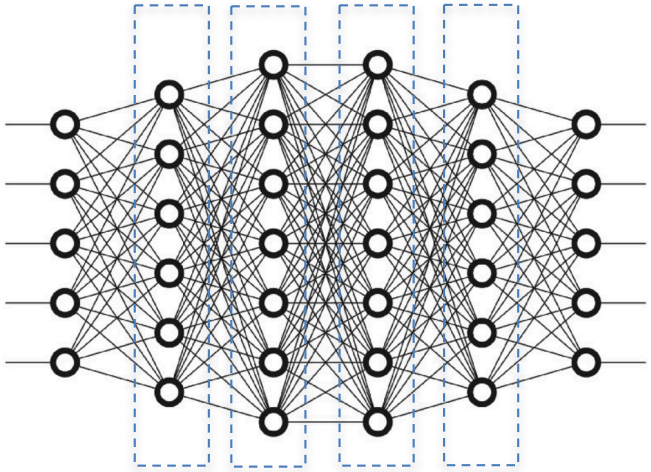
DNN architecture

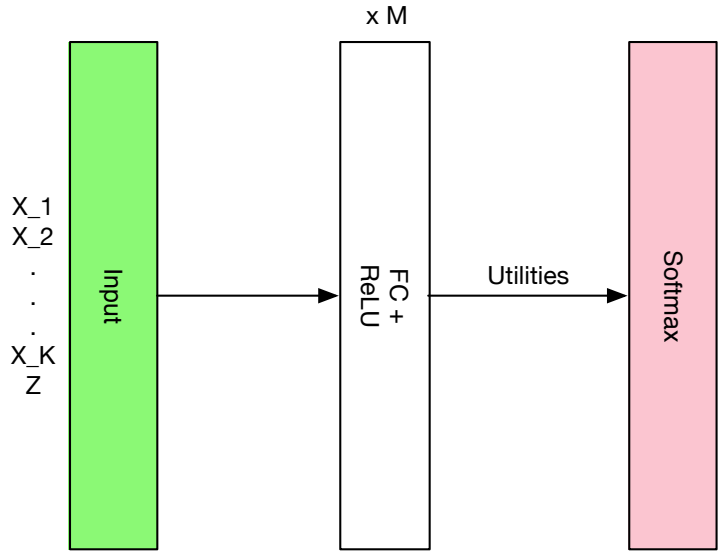
LeNet
AlexNet
ResNet
etc.

Utility specification

Multinomial Logit
Nested Logit
Mixed Logit
etc.

Utility Specification in Fully Connected DNN (F-DNN)

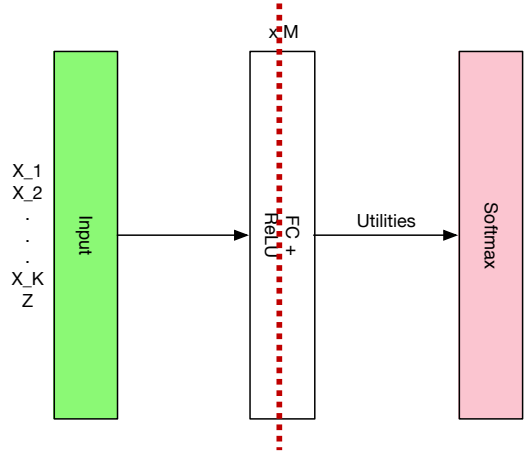




$$V_k = f(x_1, x_2, ...x_K, z)$$

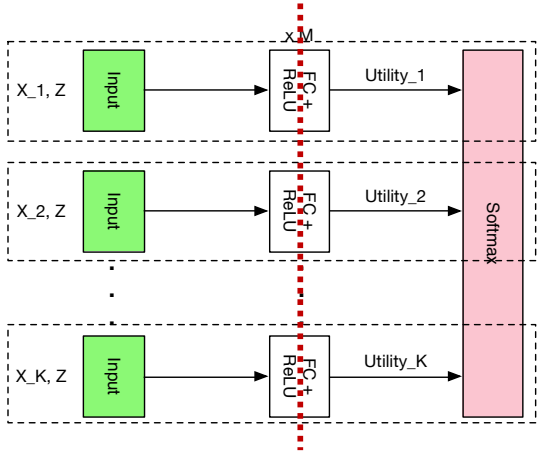
Using utility theory to design DNNs

Fully connected DNN (F-DNN)



$$V_k = f(x_1, x_2, ...x_K, z)$$

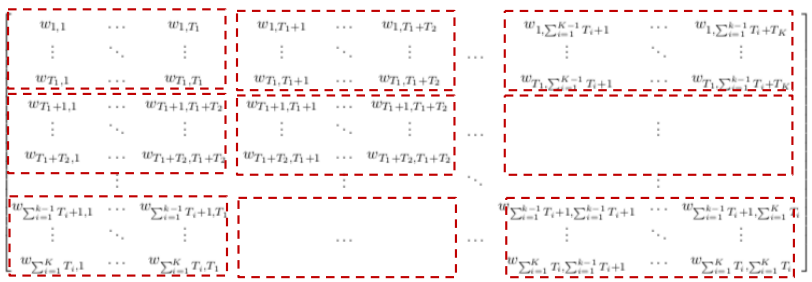
DNN with alternative-specific connection (ASU-DNN)



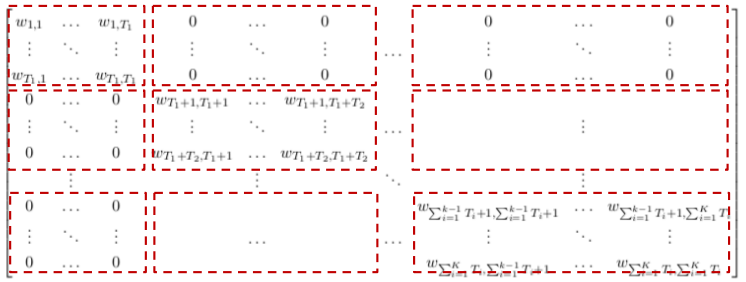
$$V_k = f(x_k, z)$$

Sparsity as regularization introduced by utility theory

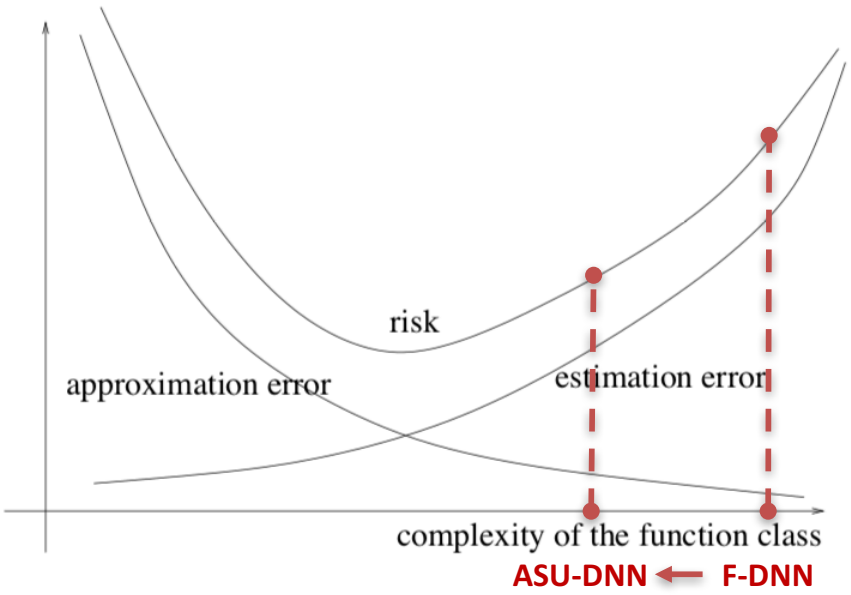
$$W_{F-DNN} =$$



$$W_{ASU-DNN} =$$



ASU-DNN: Reducing estimation errors but increasing approximation errors



Upper bound on the estimation error of **F-DNN**

$$\hat{R}_n(F_2|_S) \leq \frac{(\sqrt{2\log(D)}+1)\sqrt{\frac{1}{N}\sum_{i=1}^{N}||x_i||^2}}{\sqrt{N}} \times c^D T^D$$

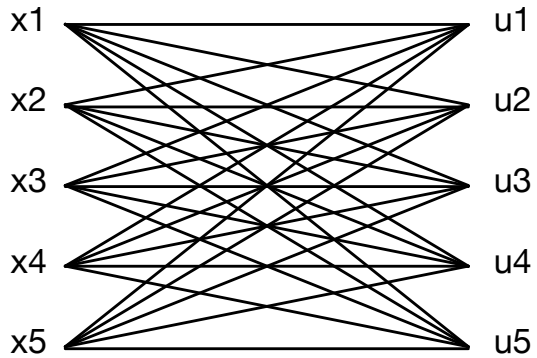
Upper bound on the estimation error of **ASU-DNN**

$$\hat{R}_n(F_1|_S) \leq \frac{(\sqrt{2\log(D)}+1)\sqrt{\frac{1}{N}\sum_{i=1}^{N}||x_i||^2}}{\sqrt{N}} \times \frac{c^D T^D}{K^{D/2}}$$

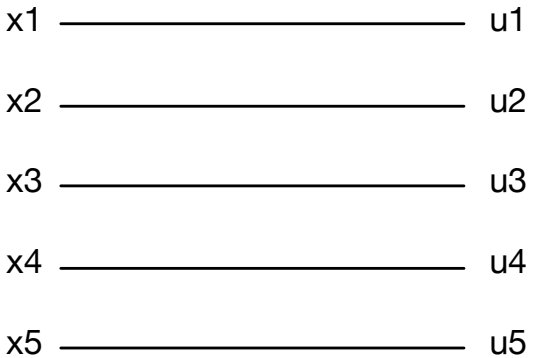
Translation: ASU-DNN becomes simpler, but the constraint might not be realistic

ASU-DNN satisfies the IIA constraint (global connectivity perspective)

F-DNN



ASU-DNN

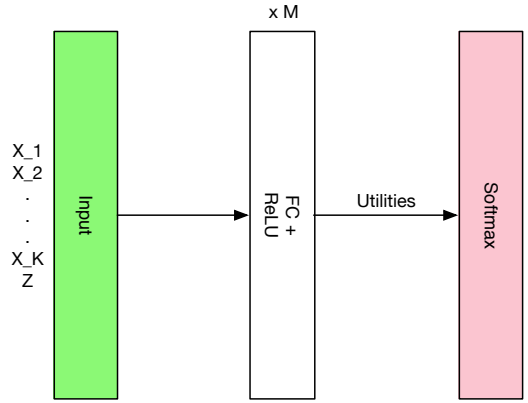


$$A_{F-DNN} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$A_{ASU-DNN} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

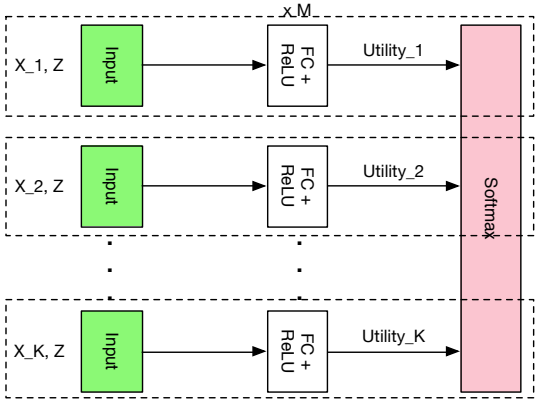
ASU-DNN satisfies the IIA constraint

Fully connected DNN (F-DNN)



$$V_k = f(x_1, x_2, ...x_K, z)$$

DNN with alternative-specific connection (ASU-DNN)



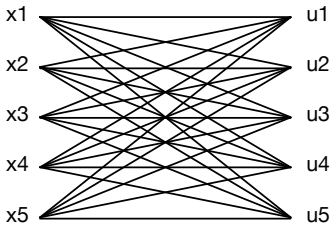
$$V_k = f(x_k, z)$$

ASU-DNN (IIA)

$$P_1/P_2 = g(x_1, x_2; z)$$

MNL ?

F-DNN



$$P_1/P_2 = g(x_1, \ldots, x_5; z)$$

MXL ?

Experiment Setup

Two datasets (SGP & TRAIN)

Sample size. SGP: 8,418; TRAIN: 2,929.

Travel mode choice (5 alternatives)

Random hyper-parameter searching

Hyperparameters	Values
Panel 1. Invariant Hyperparameters	
Activation functions	ReLU and Softmax
Loss	Cross-entropy
Initialization	He initialization
Panel 2. Varying Hyperparameters of F-DNN	
M	$[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]$
Width n	$[60, 120, 240, 360, 480, 600]$
Panel 3. Varying Hyperparameters of ASU-DNN	
M_1	$[0, 1, 2, 3, 4, 5, 6]$
M_2	$[0, 1, 2, 3, 4, 5, 6]$
Width n_1	$[10, 20, 40, 60, 80]$
Width n_2	$[10, 20, 40, 60, 80, 100]$
Panel 4. Varying Hyperparameters of F-DNN and ASU-DNN	
γ_1 (l_1 penalty)	$[1.0, 0.5, 0.1, 0.01, 10^{-3}, 10^{-5}, 10^{-10}]$
γ_2 (l_2 penalty)	$[1.0, 0.5, 0.1, 0.01, 10^{-3}, 10^{-5}, 10^{-10}]$
Dropout rate	$[0.5, 0.1, 0.01, 10^{-3}, 10^{-5}]$
Batch normalization	$[True, False]$
Learning rate	$[0.5, 0.1, 0.01, 10^{-3}, 10^{-5}]$
Num of iteration	$[500, 1000, 5000, 10000, 20000]$
Mini-batch size	$[50, 100, 200, 500, 1000]$

Evaluate Experiments

1

Prediction accuracy comparison

2

Effectiveness of IIA regularization, compared to:

(1) explicit regularizations

(2) implicit regularizations

(3) architectural hyper-parameters

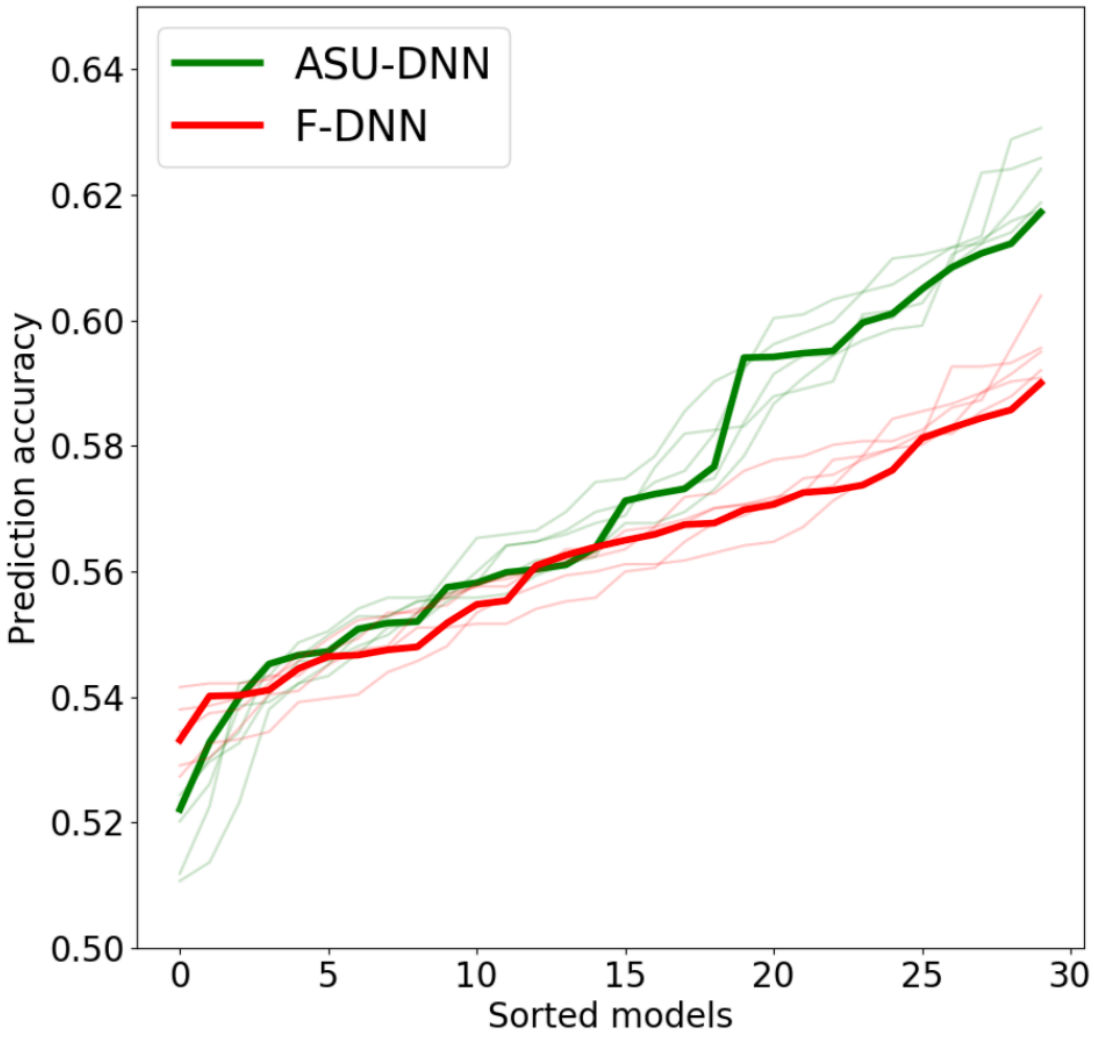
3

Interpretation: substitution patterns

Empirical Result 1: ASU-DNN outperforms F-DNN

Both validation and testing sets

Both SGP and TRAIN datasets.



Performance in Testing Set of SGP

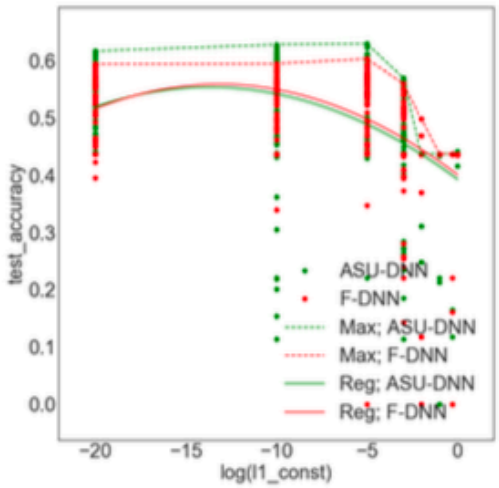
Empirical Result 1: ASU-DNN and F-DNN outperform other classifiers

	ASU-DNN (Top 1)	F-DNN (Top 1)	ASU-DNN (Top 10)	F-DNN (Top 10)	MNL (l1_reg)	MNL (l2_reg)	SVM (Linear)	SVM (RBF)	Naive Bayesian	KNN_3	DT	AdaBoost	QDA
Validation (SGP)	62.3%	59.2%	61.3%	58.8%	54.5%	54.7%	54.3%	45.6%	44.7%	58.5%	51.9%	54.6%	47.2%
Test (SGP)	61.0%	58.7%	60.4%	57.6%	52.1%	52.1%	51.8%	44.3%	41.6%	57.9%	50.2%	52.1%	44.9%
Validation (TRAIN)	70.5%	70.1%	69.8%	69.4%	69.5%	69.5%	68.8%	60.9%	57.3%	60.0%	65.0%	67.5%	60.2%
Test (TRAIN)	71.4%	72.1%	71.2%	70.7%	67.8%	67.9%	68.3%	58.7%	56.4%	57.7%	65.0%	69.8%	60.5%

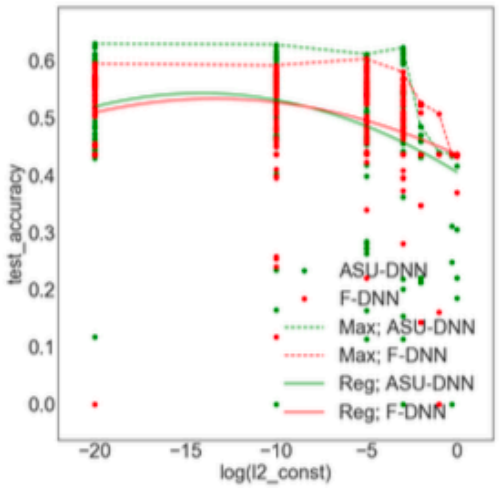
Table 1: Prediction accuracy of all classifiers; MNL (l1_reg/l2_reg) represents a multinomial logit model with mild l1 or l2 regularization; SVM (Linear/RBF) represents for support vector machine with linear or RBF kernels; KNN_3 represents three-nearest neighbor classifier; decision tree is abbreviated as DT; quadratic discriminant analysis is as QDA. The DNN models outperform all the other classifiers.

Empirical result 2: alternative-specific connection is more effective than explicit regularizations

1. Why are they called explicit regularizations?
2. Comparison



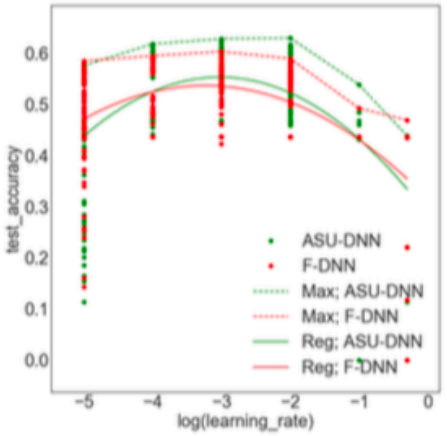
(a) l_1 Regularization



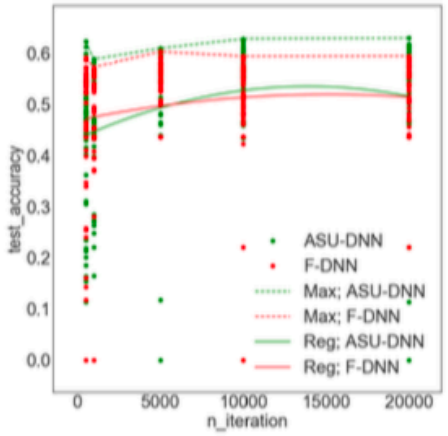
(b) l_2 Regularization

Empirical result 2: alternative-specific connection is more effective than some of the implicit regularizations

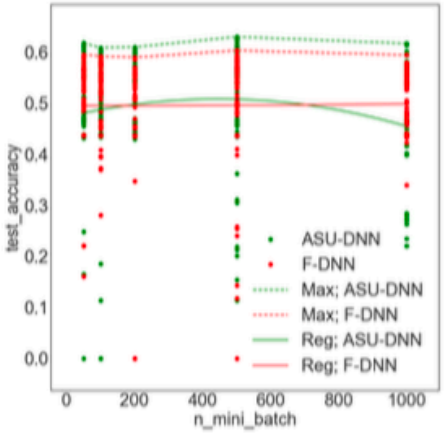
1. Why are they called implicit regularizations?
2. Comparison



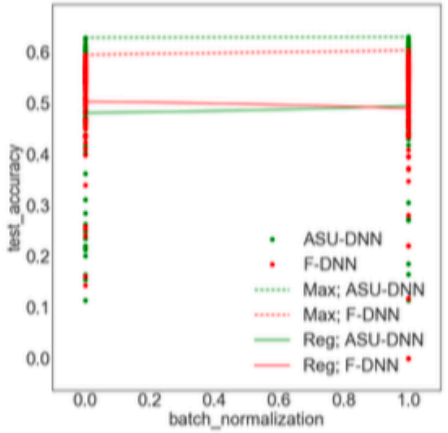
(c) Learning Rates



(d) Number of Iteration



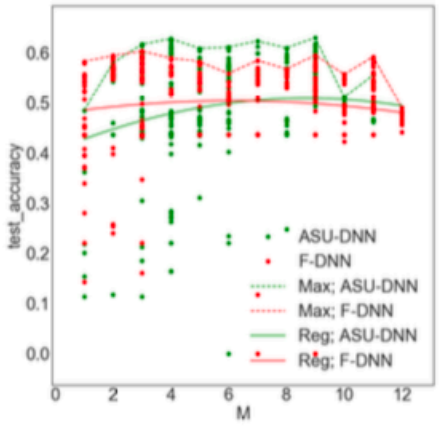
(e) Size of Mini Batch



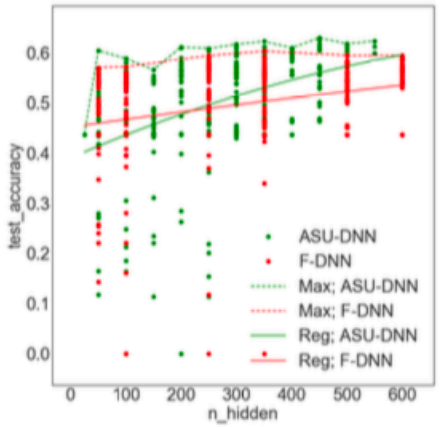
(f) Batch Normalization

Empirical result 2: alternative-specific connection is more effective than architectural hyper-parameters

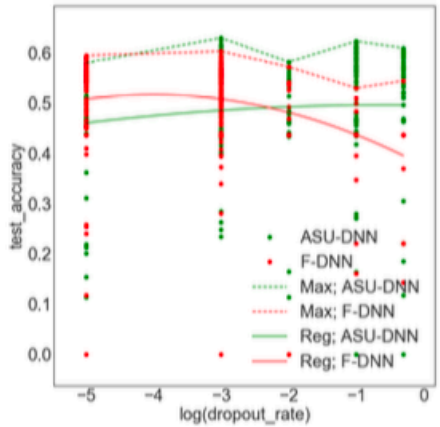
1. Why are the architectural hyper-parameters can be treated as regularizations?
2. Comparison
3. Categorization of regularizations is now clear cut.



(g) Depth of DNN



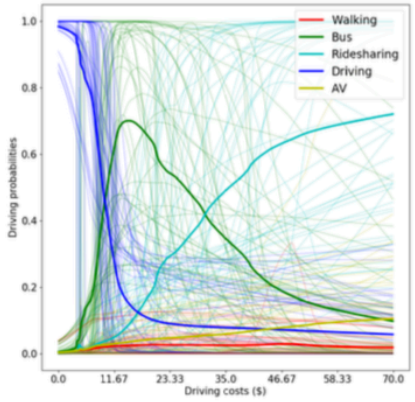
(h) Width of DNN



(i) Dropout Rates

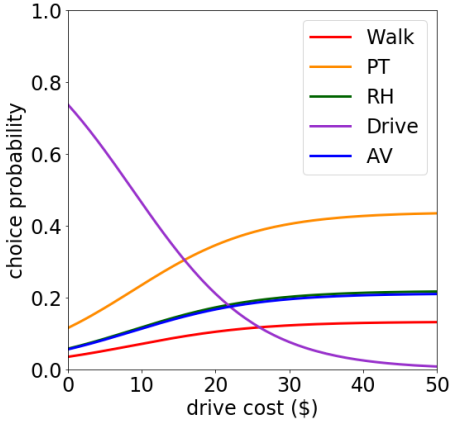
Empirical result 3: interpreting the substitution pattern

F-DNNs (maybe too complex to be true)

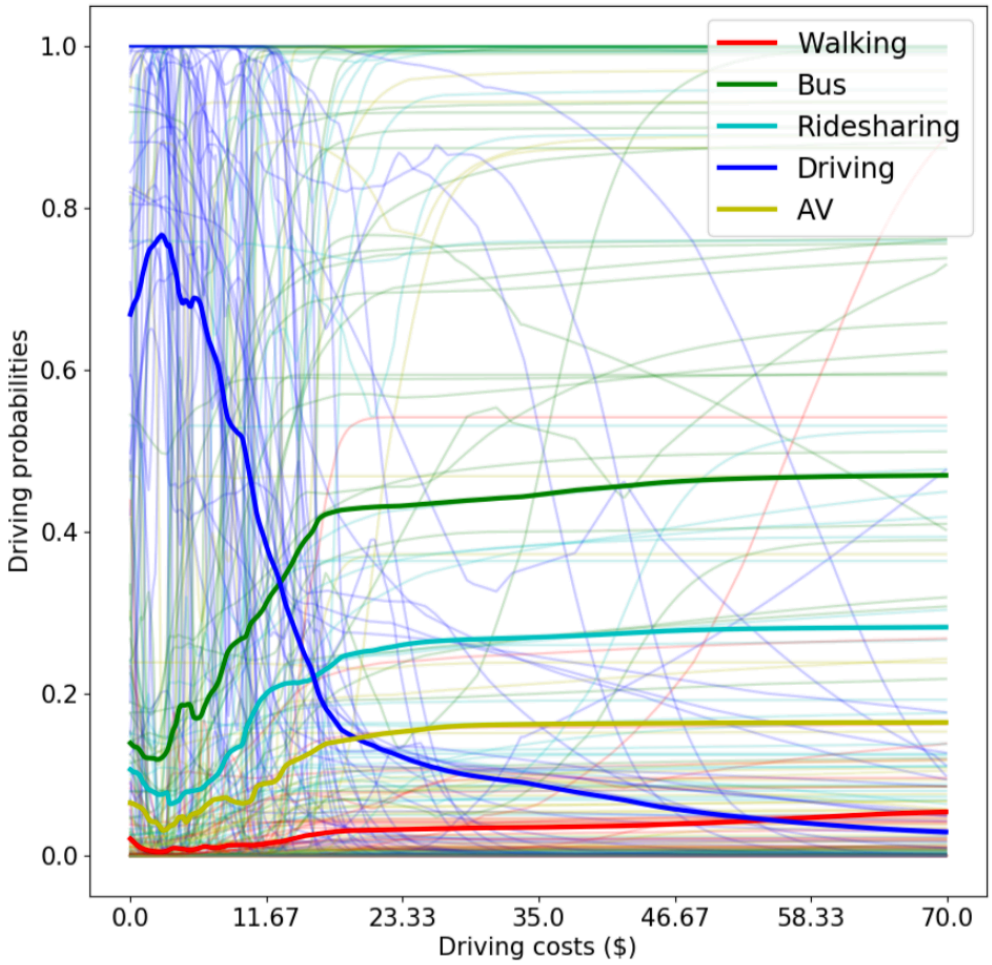


(c) F-DNN (Top 10 Models)

MNLs (IIA; maybe too simple to be true)

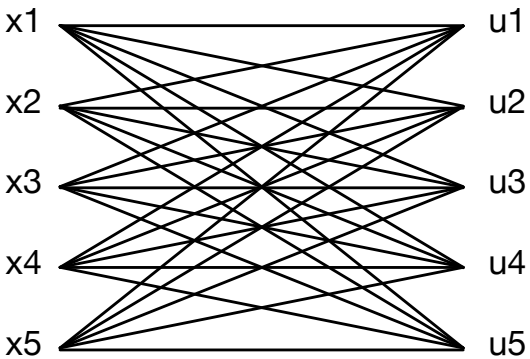


ASU-DNNs: an interesting mixture

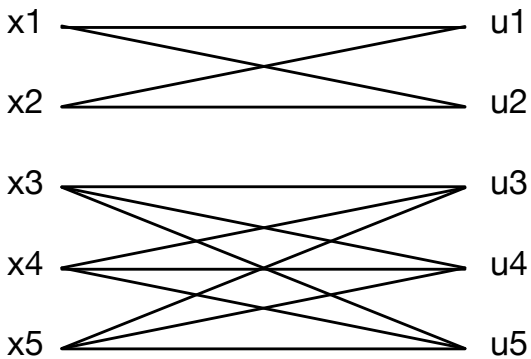


A broader perspective: DNN design with utility connectivity graph (UCG)

DNN



NSU-DNN



ASU-DNN

$$A_{F-DNN} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$A_{NSU-DNN} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

$$A_{ASU-DNN} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Substitution Patterns and IIA Constraint

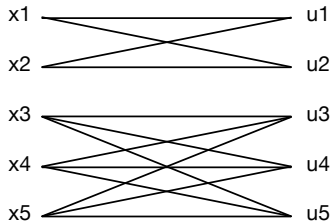
ASU-DNN (IIA)

$$P_1/P_3 = f(x_1, x_3; z)$$

$$P_1/P_2 = f(x_1, x_2; z)$$

MNL ?

NSU-DNN (IIA in Nests)

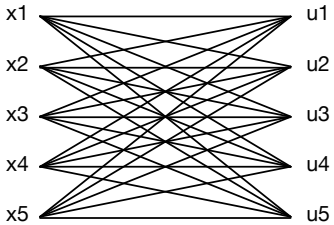


$$P_1/P_2 = f(x_1, x_2; z)$$

$$P_1/P_3 = f(x_1, x_2, ..., x_5; z)$$

NL ?

DNN



$$P_1/P_3 = f(x_1, x_2, ..., x_5; z)$$

$$P_1/P_2 = f(x_1, x_2, ..., x_5; z)$$

MXL ?

Conclusion

Model (ASU-DNN)

DNN perspective: sparsity & smaller estimation errors

Behavioral perspective: alternative-specific utility function & IIA constraint

Intersection of two perspectives: interpretable architecture by introducing behavioral knowledge into DNN

Empirical Results

Improving prediction accuracy by reducing estimation errors

Forming effective domain-knowledge-based regularization

IIA-based substitution pattern for interpretation

It is the first paper that approaches DNN interpretability for choice analysis from a behavioral perspective. (simple & neat ^_^)

Future Research Questions

What are the DNN architectures corresponding to MNL, NL, MXL, or broadly other behavioral knowledge? (Expanding the idea of UCG)

What are the utility specifications corresponding to LeNet, AlexNet, and GoogLeNet?