How to Integrate Deep Learning Methods with Transportation Model Calibration: A Computational Graph-Based Approach with Multiple Data Sources

Xuesong (Simon) Zhou, xzhou74@asu.edu

School of Sustainable Engineering and the Built Environment
Arizona State University

Co-author: Dr. Xin (Bruce) Wu
xinwu3@asu.edu
Postdoc scholar at ASU

Arizona State University

Contents

1. Challenges in planning modal calibration with **multiple data sources**

2. What is **Deep Learning** from our model calibration perspective?

3. What is **layered computation graph** anyway?

4. Steps for **integrating ML/CG in 4-step** model calibration

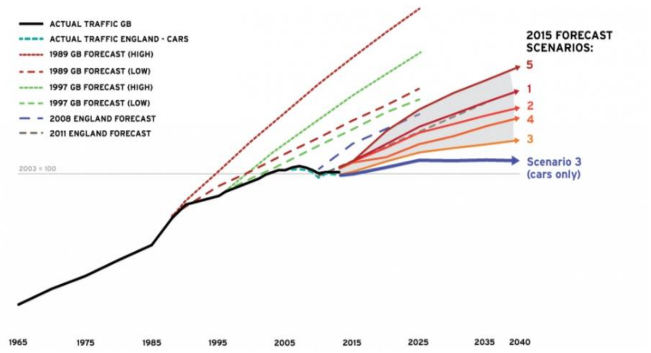
5. Simultaneous **Forecasting** Model Using an Econometric model +Deep Neural Network

6. Further extensions

Arizona State University

1.Challenges in planning modal calibration with multiple data sources

Introduction

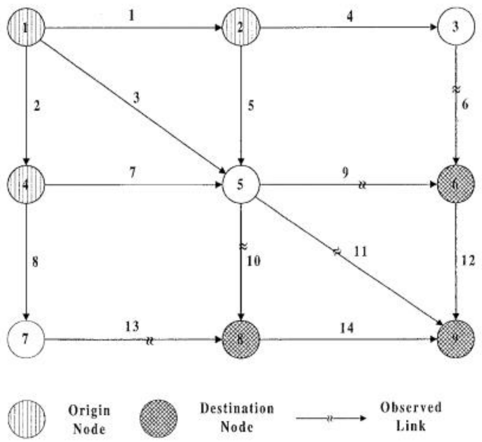


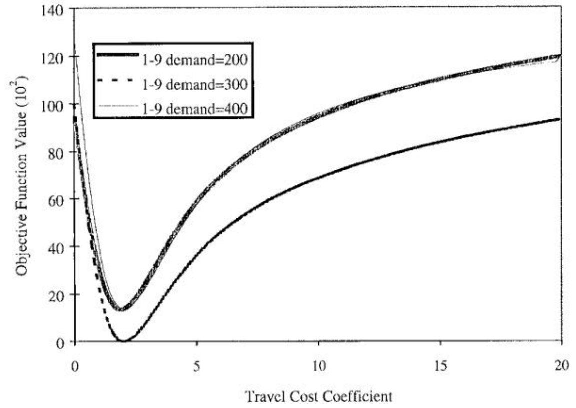
- **Basic concepts**
- 1. Traffic demand flow estimation (TDFE) problem
- 2. The simultaneous estimation problem of traffic demand flows and users behavior coefficients (TDFE-UB)
  - **Trip generation**→ Linear regression model given OD split rates and link proportion
  - **OD Matrix** →ODME problem, inverse process of four-step models
  - **Path/Link flows**→ Traffic assignment/ Stochastic network loading
  - **Behaviors parameters** (e.g. Value of time, VOT) → Discrete choice model

Simultaneous Estimation of the Origin-Destination Matrices and Travel-Cost Coefficient for Congested Networks in a Stochastic User Equilibrium

Hai Yang • Qiang Meng • Michael G. H. Bell
Department of Civil Engineering, The Hong Kong University of Science & Technology, Clear Water Bay, Kowloon, Hong Kong, China
Department of Civil Engineering, The Hong Kong University of Science & Technology, Clear Water Bay, Kowloon, Hong Kong, China
Department of Civil Engineering, The University of Newcastle upon Tyne, NE1 7RU, United Kingdom

Transportation Science; May 2001; 35, 2; ABI/INFORM Collection
pg. 107





Problem is **nonconvex**

ods. In this respect, very little can be done theoretically due to the high nonlinearity of the model and extensive numerical simulation experiments have to be employed. Furthermore, because of the inherent nonconvex property of the problem, development of some global optimization techniques such as a genetic algorithm or simulated annealing would be highly desirable.

Introduction

- **Big data resources:**

(1) Household travel surveys

- Empirical foundation of many classical models
- Expensive and the period of surveys range from 5 to 10 years
- Demographic characteristics

(2) Mobile phone data

- Demographic biases (e.g. income or age bias)
- The spatial granularity of phone data depends on the cell tower to which call records are mapped (often >100m, about 200m-2000m) .
- Do not know traffic mode
- Map matching process is limited by locational precision of cell-towers.
- Sample penetration of the data

Introduction

- **Big data resources:**

(3) Floating car data

- Higher resolution than mobile phone data (time-dependent map matching)
- Taxi and bus only
- Low sample penetration of the data

(4) Sensor data

- Underdeterminedness
- Some links have not yet installed sensors.
- Overfitting

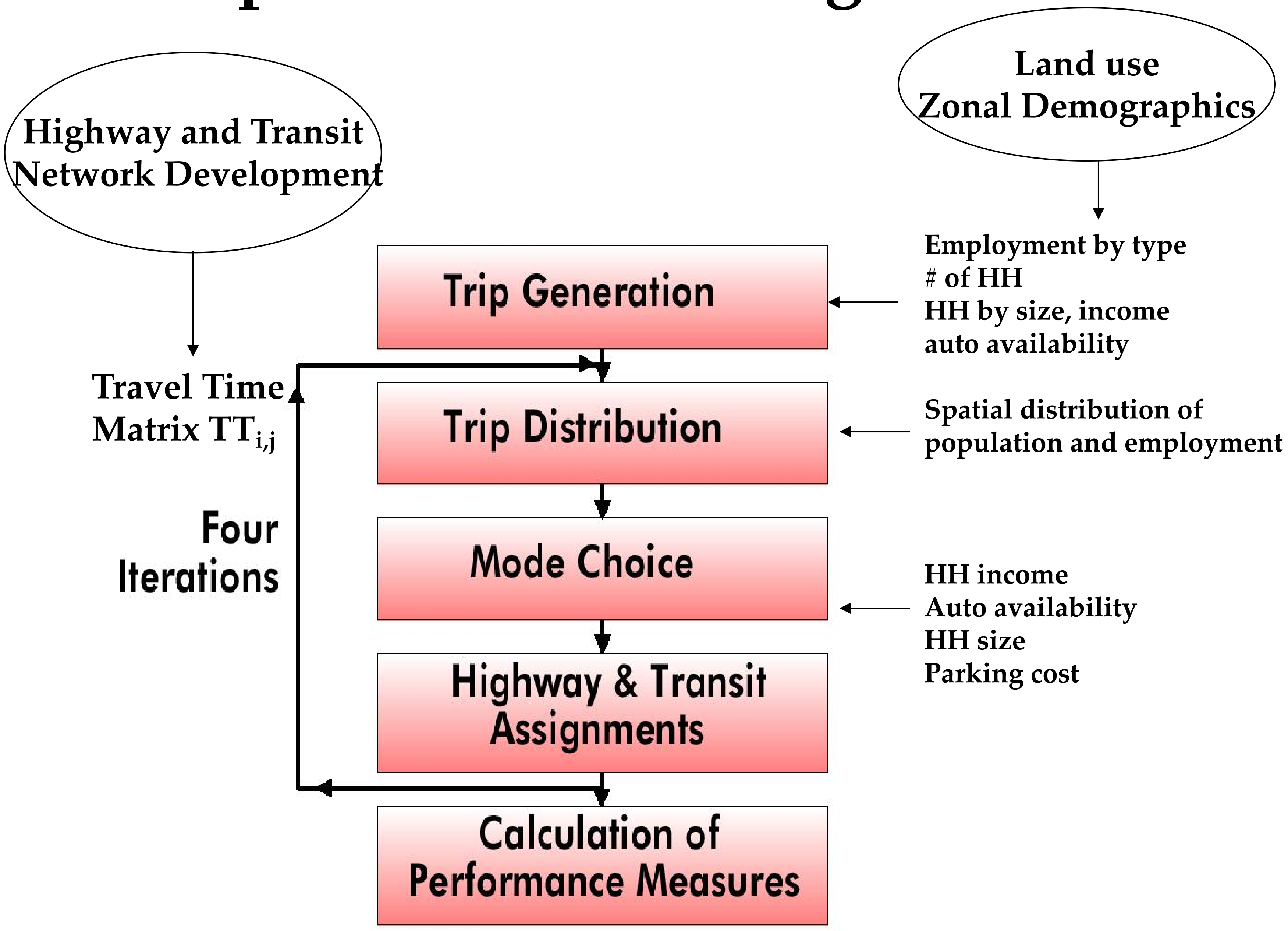
Introduction

Big data resources:

Comparison of different data resources.

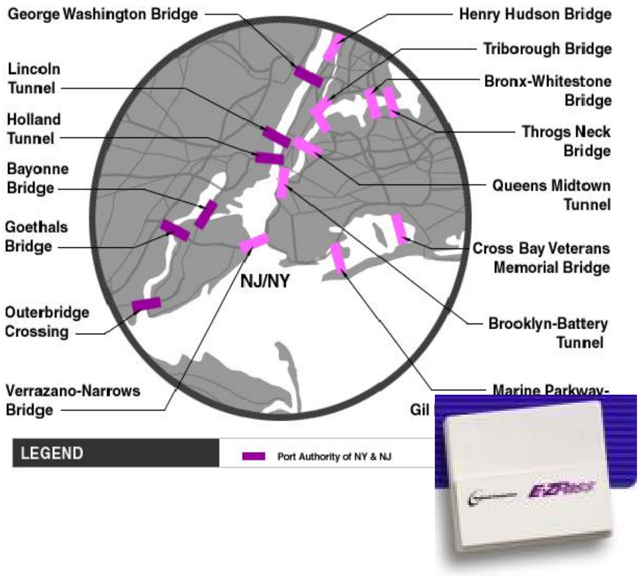
Characteristics	Household travel surveys	Mobile phone data	Floating car data	Sensor data
Time period	5-10 years	Multi hour periods within the day	individual hours of the day	Per 15 minutes of the day
Demand Types	Aggregated	Aggregated	Aggregated sometime disaggregated traces	Aggregated
Sample penetration	Low	Depend on market share	Low	High
Precision	Zone-based	200m-2000m	1-10m	Link-based
Coverage Issues	High coverage	Coverage-limited	Coverage-limited	Coverage-limited
Demographic Bias	Affluent demographic information	Mild demographic bias	Severe demographic bias	Do not have demographic bias for car owners
Speed information	Not available	Not available	Available with varying degrees of processing effort	Generally available
Link/Corridor analysis	Not available	Not available	Limited available	Available
Data collecting cost	Expensive	Inexpensive	Inexpensive	Inexpensive

4-step process in transportation modeling

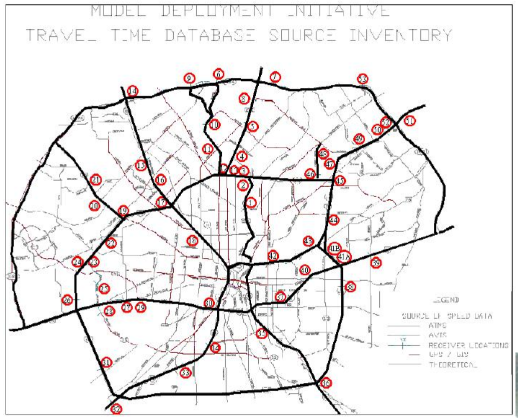


Additional Data Source 1: Automatic Vehicle Identification Data

Toll Facilities in the New York City Area



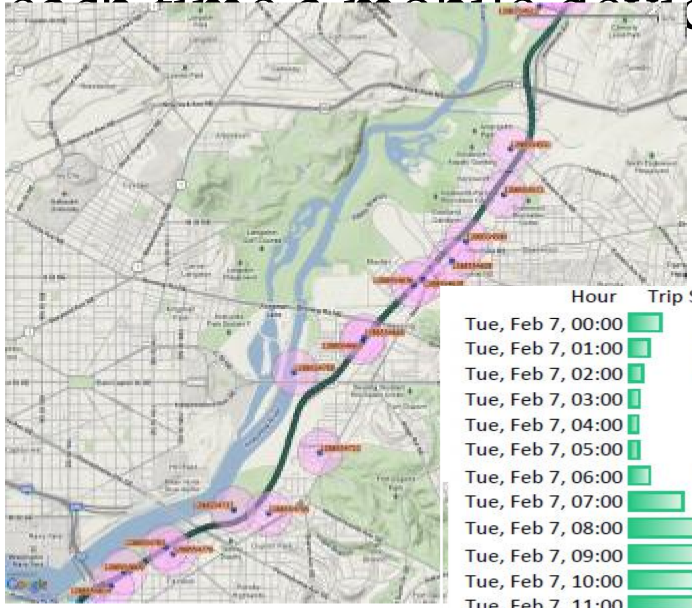
AVI Readers in San Antonio, TX



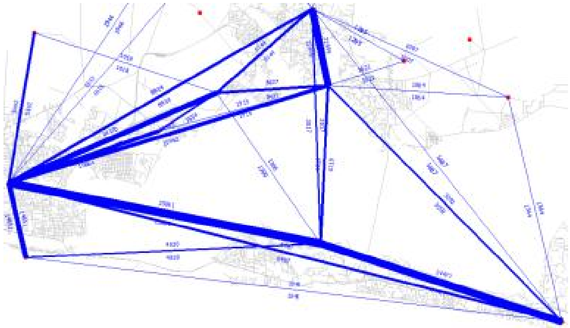
	Identification rates	Market penetration rates
Active RFID tags	>99.9%	Low
Passive RFID tags	75%-85%	Low
License plates	40%-65%	100%

Additional Data Source 2: Cell Phone Data: e.g. Airsage

- AirSage processes and archives a location each time a mobile device interacts with the network…
  - Start of Call
  - End of Call
  - Text Message
  - Data Transfers
  - Multiple hourly location update
  - **Aggregate Population Movements (OD)**



Hour	Trip Starts	Trip Ends	Daily %
Tue, Feb 7, 00:00	2216	1578	2.2%
Tue, Feb 7, 01:00	1405	1241	1.4%
Tue, Feb 7, 02:00	1004	934	1.0%
Tue, Feb 7, 03:00	751	707	0.7%
Tue, Feb 7, 04:00	682	601	0.7%
Tue, Feb 7, 05:00	712	555	0.7%
Tue, Feb 7, 06:00	1455	808	1.4%
Tue, Feb 7, 07:00	3662	2626	3.6%
Tue, Feb 7, 08:00	5225	4357	5.1%
Tue, Feb 7, 09:00	5186	4882	5.1%
Tue, Feb 7, 10:00	4477	4391	4.4%
Tue, Feb 7, 11:00	5898	5739	5.8%
Tue, Feb 7, 12:00	6471	6269	6.3%
Tue, Feb 7, 13:00	7943	7728	7.7%
Tue, Feb 7, 14:00	5504	5388	5.4%
Tue, Feb 7, 15:00	6502	6480	6.3%
Tue, Feb 7, 16:00	6677	6765	6.5%
Tue, Feb 7, 17:00	7187	7481	7.0%
Tue, Feb 7, 18:00	7911	8730	7.7%
Tue, Feb 7, 19:00	4633	5794	4.5%
Tue, Feb 7, 20:00	4011	4850	3.9%
Tue, Feb 7, 21:00	4347	4575	4.2%
Tue, Feb 7, 22:00	4599	4744	4.5%
Tue, Feb 7, 23:00	4104	3549	4.0%



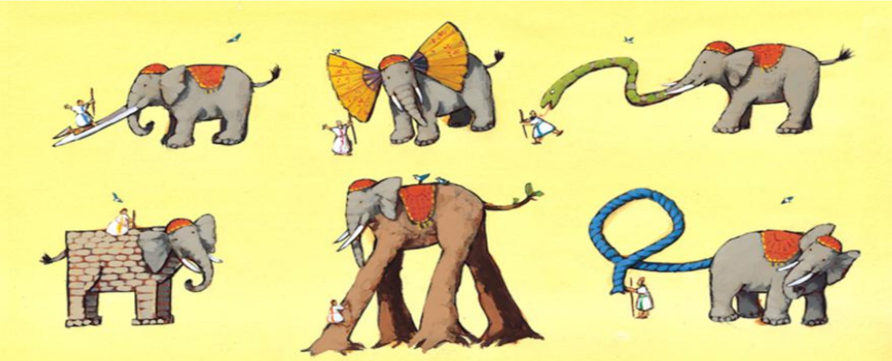
Do we still need analytical model?

- Integrate domain knowledge with different traffic measurements

Automatic Vehicle Location

Household Surveys

Cell phone data

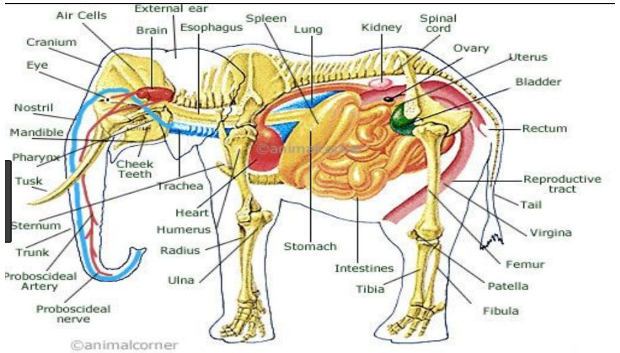


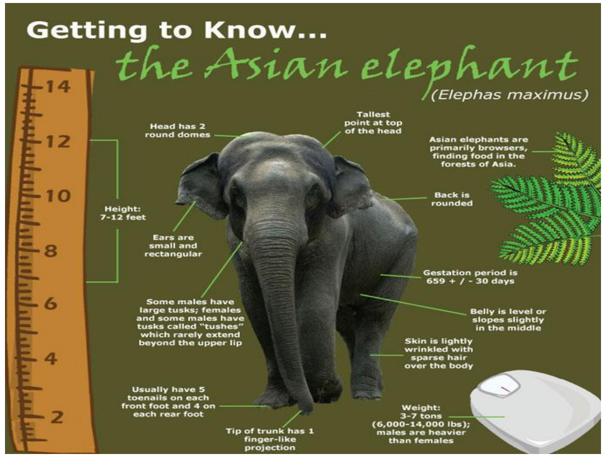
Loop detector data

Smart card data

Social location data

Domain knowledge from transportation modeling



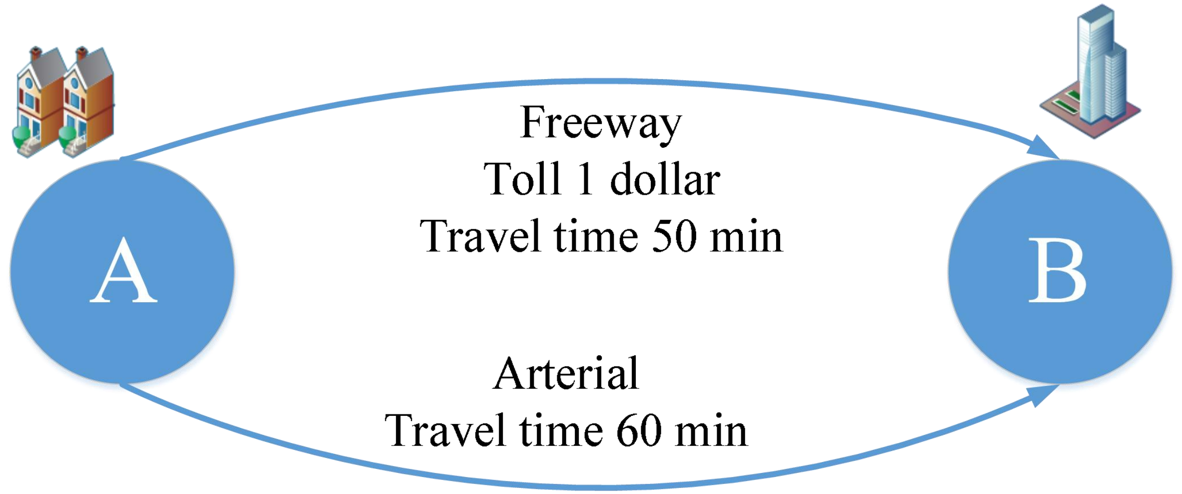


Network state of a real transportation system

Why we integrate different types of data sources?

Residential Survey
3000 households; trip rate: 3.5
Number of trips 10,500
VOT: 20 dollar/hour
Sensor
Freeway: 6,500

Cell phone
Total number of records: 15,000
Split: Freeway 65%; Arterial 35%



Why we integrate different types of data sources?

Which one is right?

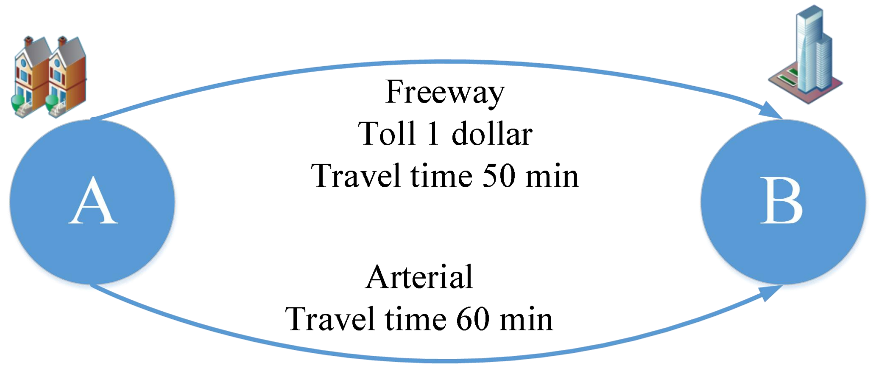
Scenario 1 : only use cell phone data

Total number: 15000 trips

Split: Freeway 65%; Arterial 35%

Flows on freeway: 9750 trips

Flows on arterial: 5250 trips



Scenario 2 : only use sensor data

Flows on freeway: 6500 trips

(overfitting)

Scenario 3: only use residential survey

Total number 10500 trips VOT=20 dollar/hour

$$10500 \times \frac{\exp\{-20 \times 0.833 - 1\}}{\exp\{-20 \times 0.833 - 1\} + \exp\{-20 \times 1\}} = 9577$$

$$10500 \times \frac{\exp\{-20 \times 1\}}{\exp\{-20 \times 0.833 - 1\} + \exp\{-20 \times 1\}} = 923$$

Flows on freeway: 9577 trips

Flows on arterial: 923 trips

Why we integrate different types of data sources?

Comparison of different data sources

Characteristics	Household travel surveys	Mobile phone data	Automatic Vehicle Location data	Loop detector data
Time period	5-10 years	Multi-hour periods within the day	individual hours of the day	Per 15 minutes of the day
Demand types	Aggregated and disaggregated	Aggregated	Aggregated sometime disaggregated traces	Aggregated
Sample penetration	Low	Depend on market share	Low	High
Spatial resolution	Household-based	200m-2000m	1-10m	Link-based
Spatial coverage	Zone-based, individuals	origin to destination	OD and path	Link

Wu, X., Guo, J., Xian, K., & Zhou, X. (2018). Hierarchical travel demand estimation using multiple data sources: A forward and backward propagation algorithmic framework on a layered computational graph. *Transportation Research Part C: Emerging Technologies*, *96*, 321-346.

Is a combined analytical 4-step model easy to calibrate?

Traffic demand flow estimation (TDFE) problem (Simultaneous estimation problem of traffic demand flows and behavior coefficients)

Data sources

Traditional Survey
Big data sources

Different levels of demand variables

1. Trip generation
2. Spatial distribution
3. Route choices
4. Behavioral utilities

Sequential step methods:

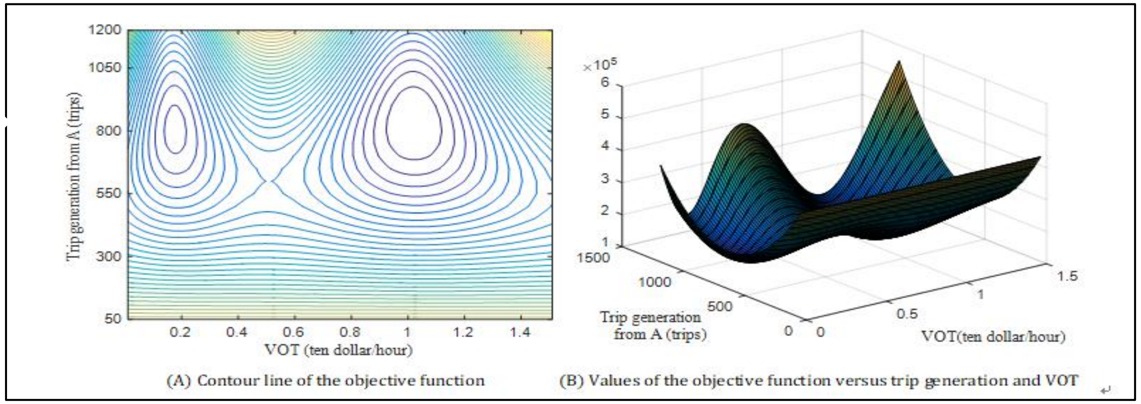
1. Linear regression model; Fundamental units
2. Gravity model; ODME problem
3. Traffic assignment/ Stochastic network loading
4. Discrete choice model

A. Usually only use single data sources for each step
B. Sequential decisions without feedback


Simultaneous Estimation of the Origin-Destination Matrices and Travel-Cost Coefficient for Congested Networks in a Stochastic User Equilibrium

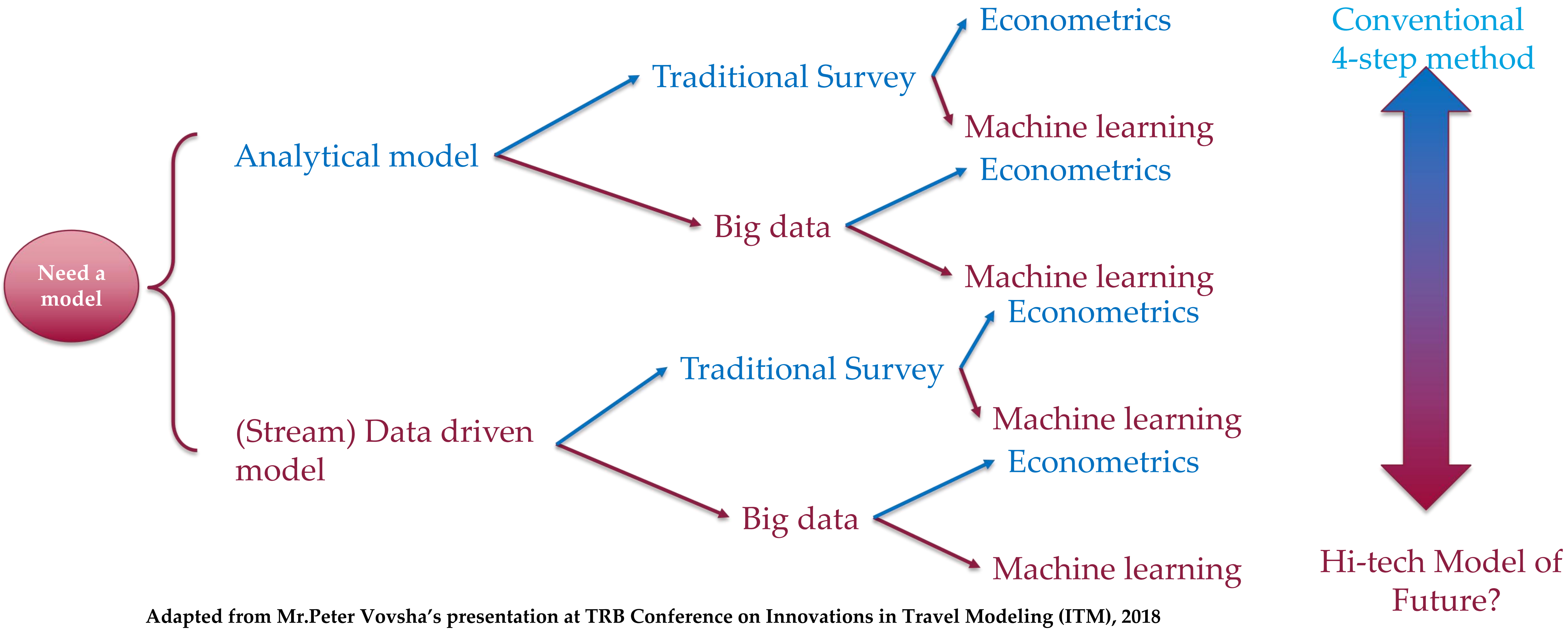
Hai Yang • Qiang Meng • Michael G. H. Bell
Department of Civil Engineering, The Hong Kong University of Science & Technology, Clear Water Bay, Kowloon, Hong Kong, China
Department of Civil Engineering, The Hong Kong University of Science & Technology, Clear Water Bay, Kowloon, Hong Kong, China
Department of Civil Engineering, The University of Newcastle upon Tyne, NE1 7RU, United Kingdom

- **Non convex**
- **Hard to solve**



(A) Contour line of the objective function
(B) Values of the objective function versus trip generation and VOT

What kinds of models do we need in the future?



Adapted from Mr.Peter Vovsha's presentation at TRB Conference on Innovations in Travel Modeling (ITM), 2018

1. Do we need a pure analytical model or a data driven model, or a data-driven analytical model?
2. How big data with different data sources, in conjunction with traditional survey data, can be used to calibrate conventional 4-step model?
3. What is the relationship between classical econometric model and machine learning models? Can machine learning models be used to calibrate 4-step model?

2. What is Deep Learning from our model calibration perspective?

What can we learn from AI field?

1. Hierarchical structure of artificial neural networks

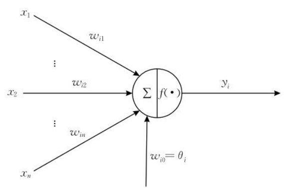
Brief history of deep learning

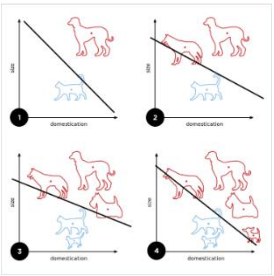
McCulloch-Pitts (1943)
Neuron of brain function

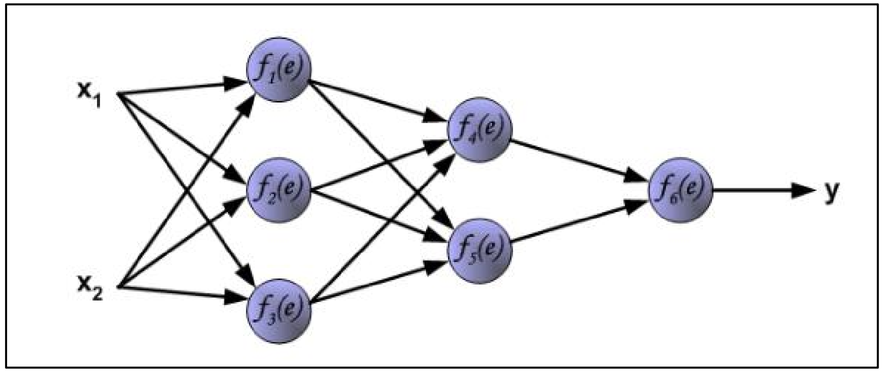
Rosenblatt (1956)
Perceptron ADA-LINE
Stochastic gradient descent

Rumelhart et al. (1986)
Multi-layer distributed representation
Back propagation algorithm

Hinton et al. (2006)
The last ten years
Convolutional networks







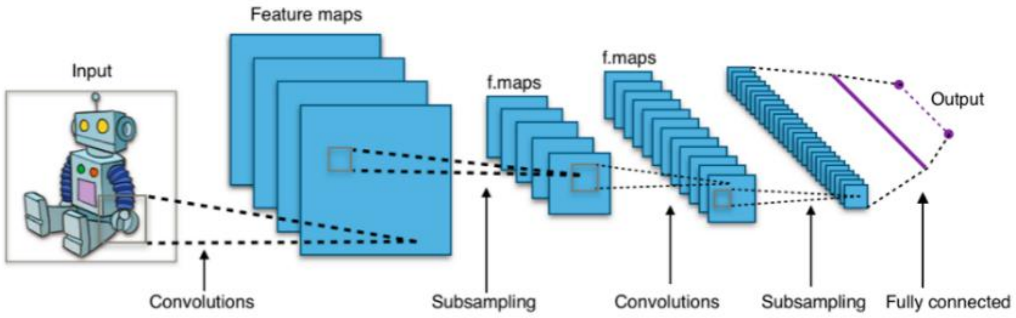
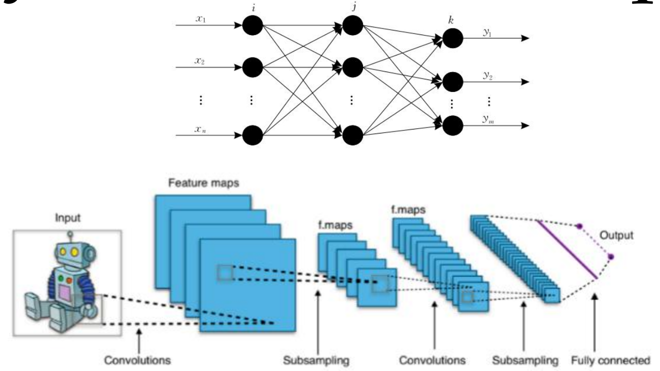
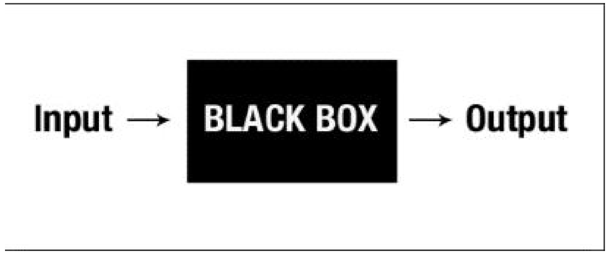


Figure source: http://galaxy.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html

What can we learn from AI field?

Why cannot use deep learning directly?





Do not have Interpretability

Ground truth

3000 households; 3.333 trips/h

Number of trips: 10,000

66.15%: 6,615 trips on the freeway

33.85%: 3,385 trips on the arterial

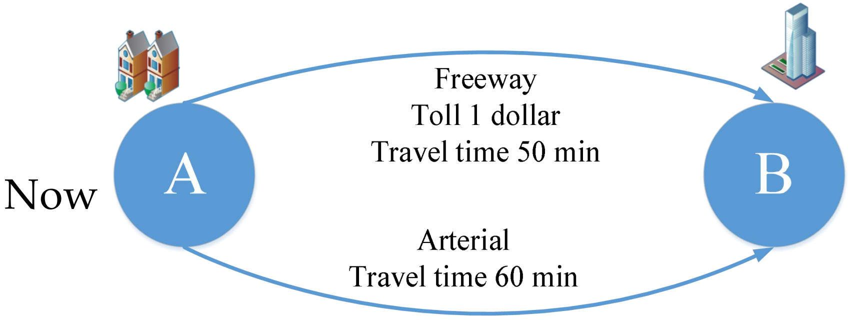
Current toll: 1 dollar

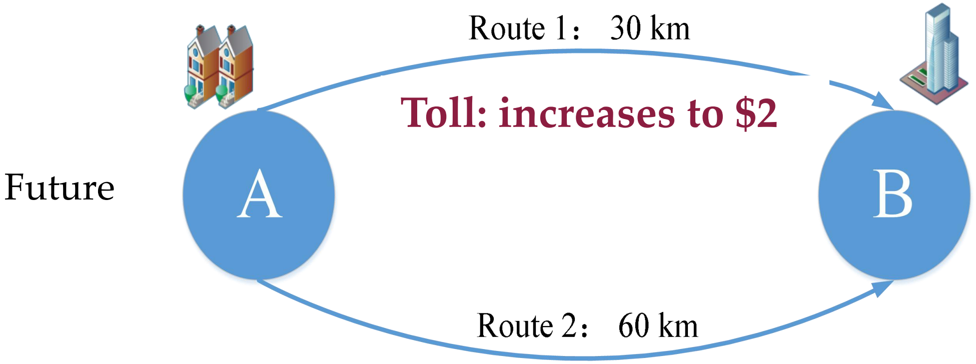
VOT:10 dollar/h

Real-world travel time of freeway: 50 min

Real-world travel time of arterial: 60 min

If the toll increases to 2 dollars, how the demand will changes?

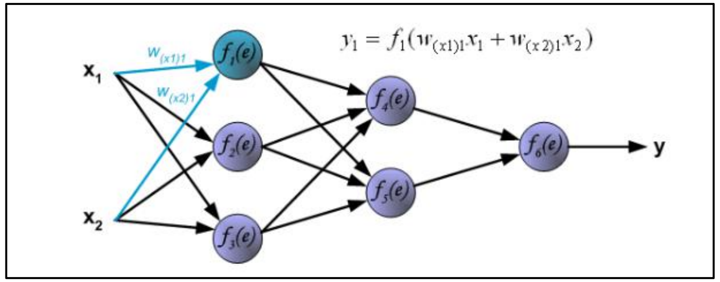


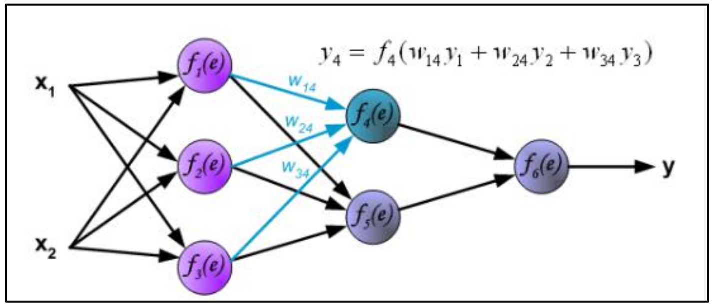


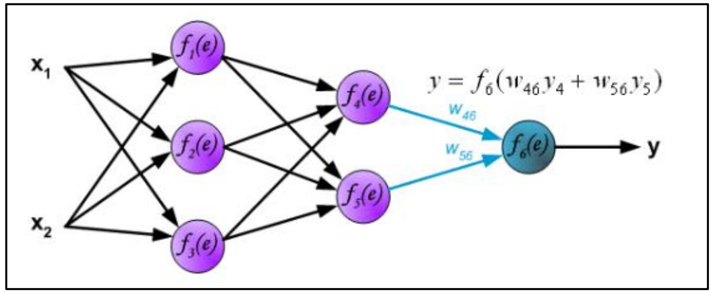
What can we learn from AI field?

2. Forward and Backward propagation

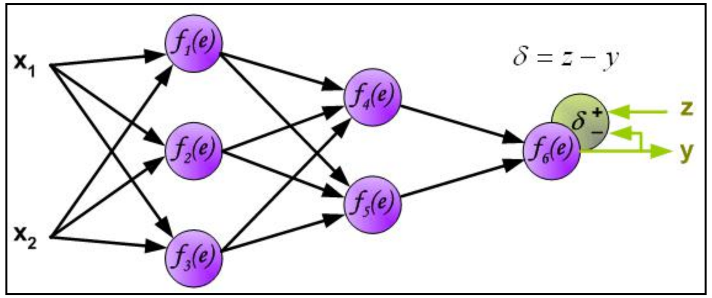
Forward propagation

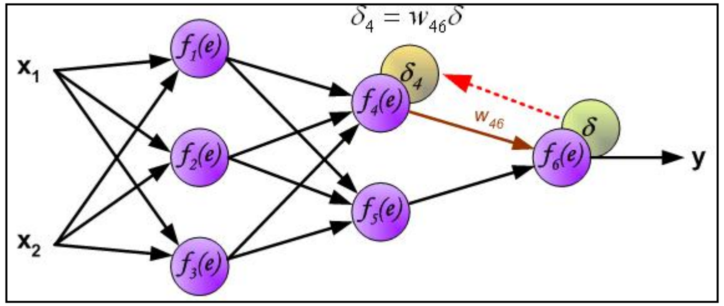


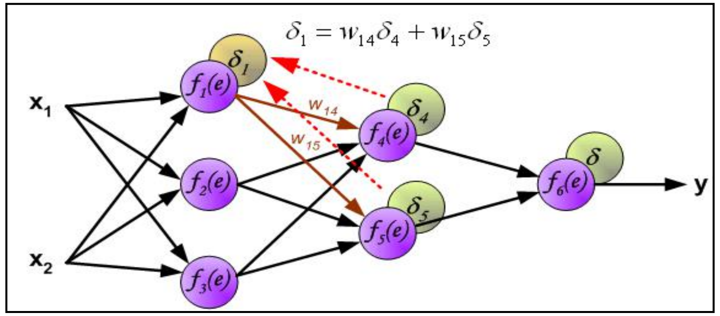




Backward propagation







Arizona State University

Source: http://galaxy.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html

What can we learn from AI field?

3. Math foundation of neural network architecture:

Computational graph approach

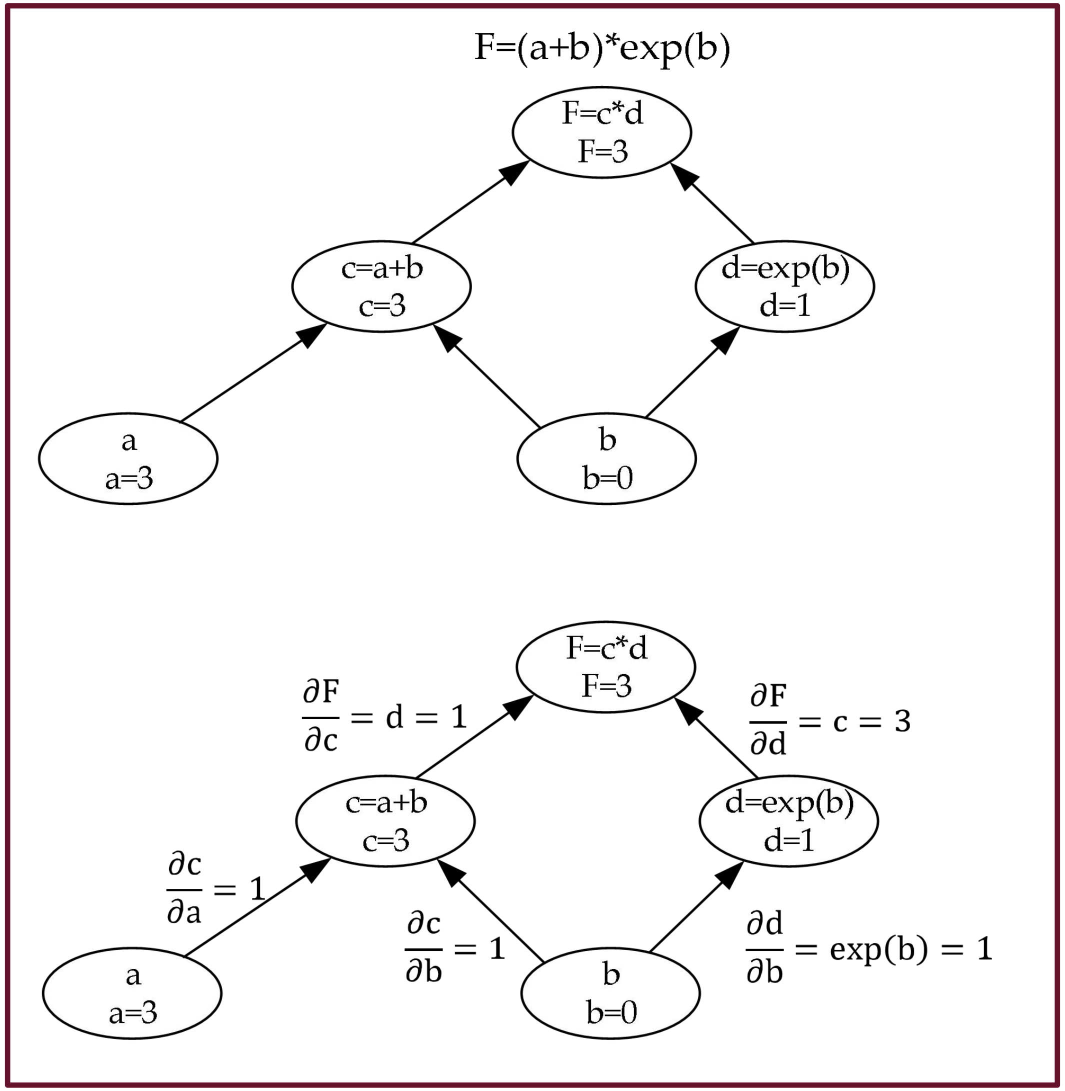
- Computational graph is a acyclic graph to express composite mathematical formulations
  (A generalization of Artificial Neural Network)

- Computational graph is a technique for calculating derivatives quickly
  (A generalization of Back propagation algorithm)

To evaluate the partial derivatives in this graph, we just need to "summing over the paths". For example, to get the derivative of **F** with respect to **b** by:

$$\frac{\partial F}{\partial b} = \frac{\partial F}{\partial c}\frac{\partial c}{\partial b} + \frac{\partial F}{\partial d}\frac{\partial d}{\partial b} = 1 * 1 + 3 * 1 = 4$$

$$\frac{\partial F}{\partial a} = \frac{\partial F}{\partial c}\frac{\partial c}{\partial a} = 1 * 1 = 1$$



Adapted from ：http://colah.github.io/posts/2015-08-Backprop/

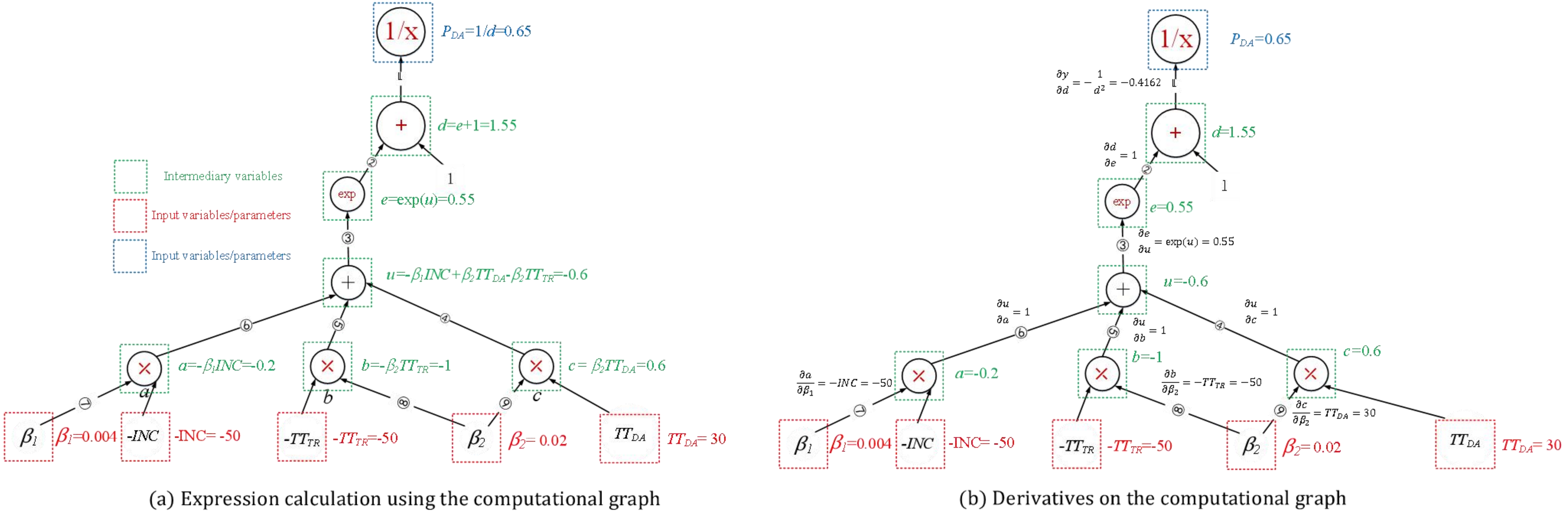
What can we learn from AI field?

3. One foundation for Deep Learning: Computational graph

Mode split models (driving alone vs. transit service) **CAN ALSO BE EXPRESSED** by a computational graph

$$P_{DA} = \sigma(U_{DA} - U_{TR}) = \frac{1}{1+\exp(U_{TR} - U_{DA})} = \frac{1}{1+\exp(-\beta_1 INC + \beta_2 TT_{DA} - \beta_2 TT_{TR})}$$

$$\frac{\partial P_{DA}}{\partial \beta_2} = \frac{\partial P_{DA}}{\partial d}\frac{\partial d}{\partial e}\frac{\partial e}{\partial u}\frac{\partial u}{\partial b}\frac{\partial b}{\partial \beta_2} + \frac{\partial P_{DA}}{\partial d}\frac{\partial d}{\partial e}\frac{\partial e}{\partial u}\frac{\partial u}{\partial c}\frac{\partial c}{\partial \beta_2} = (TT_{TR} - TT_{DA})\,\sigma(U_{DA} - U_{TR})[1 - \sigma(U_{DA} - U_{TR})]$$



What kinds of DL models do we need in model calibration?

1. Simultaneous models to **estimate different levels of traffic demands** (trip generation, trip distribution, route choices etc.)

2. Data driven models that can **integrate different data sources.**

3. **Explainable models** that combine emerging AI technologies with domain knowledge in transportation modeling.

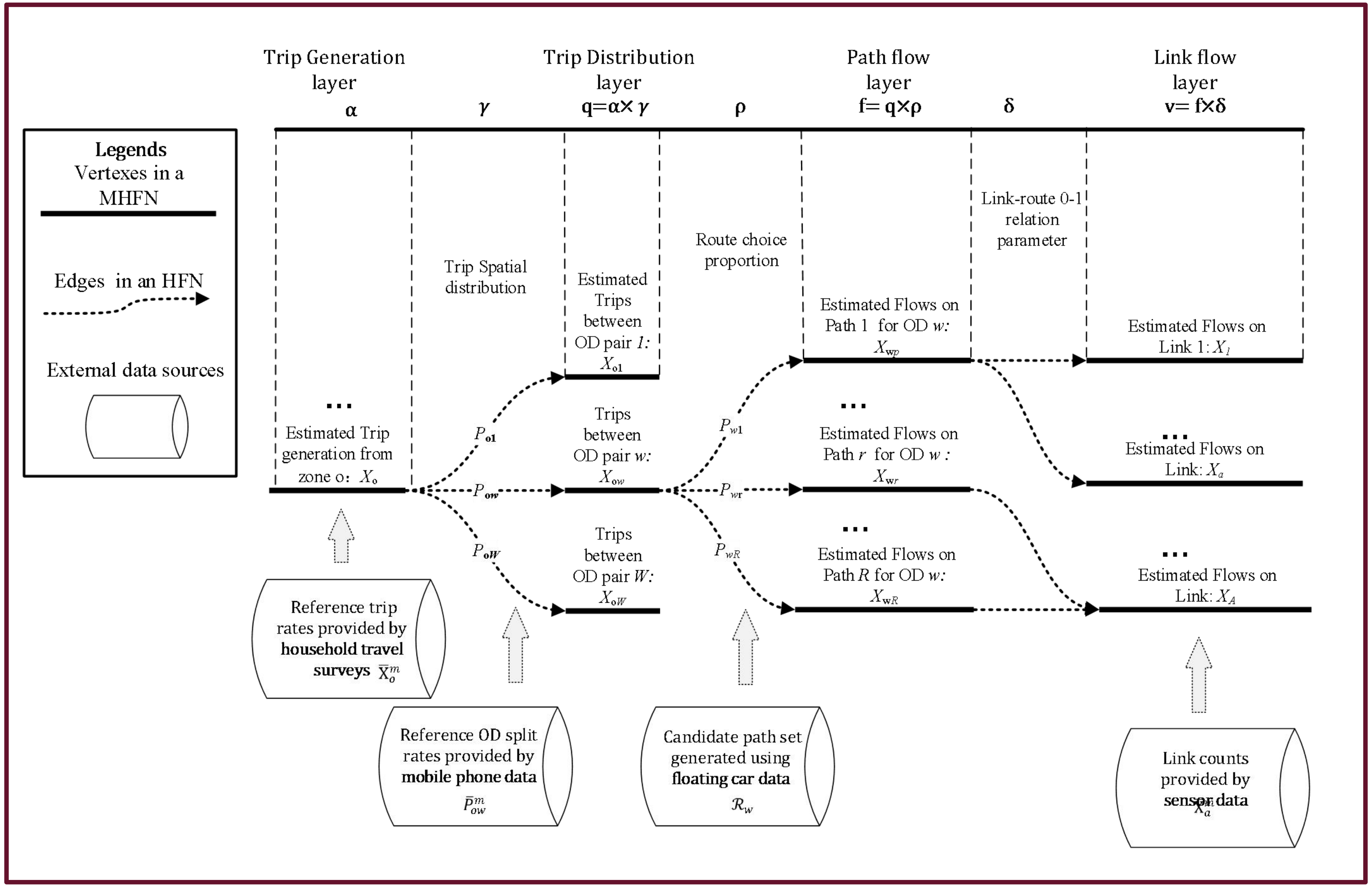
4. An iterative framework includes **both a feed forward and feedback process.**

3.What is layered computation graph?

A forward and backward propagation algorithmic framework on a layered computational graph

3.1 Hierarchical Flow Network (HFN) representation

HFN representation in a explicit form

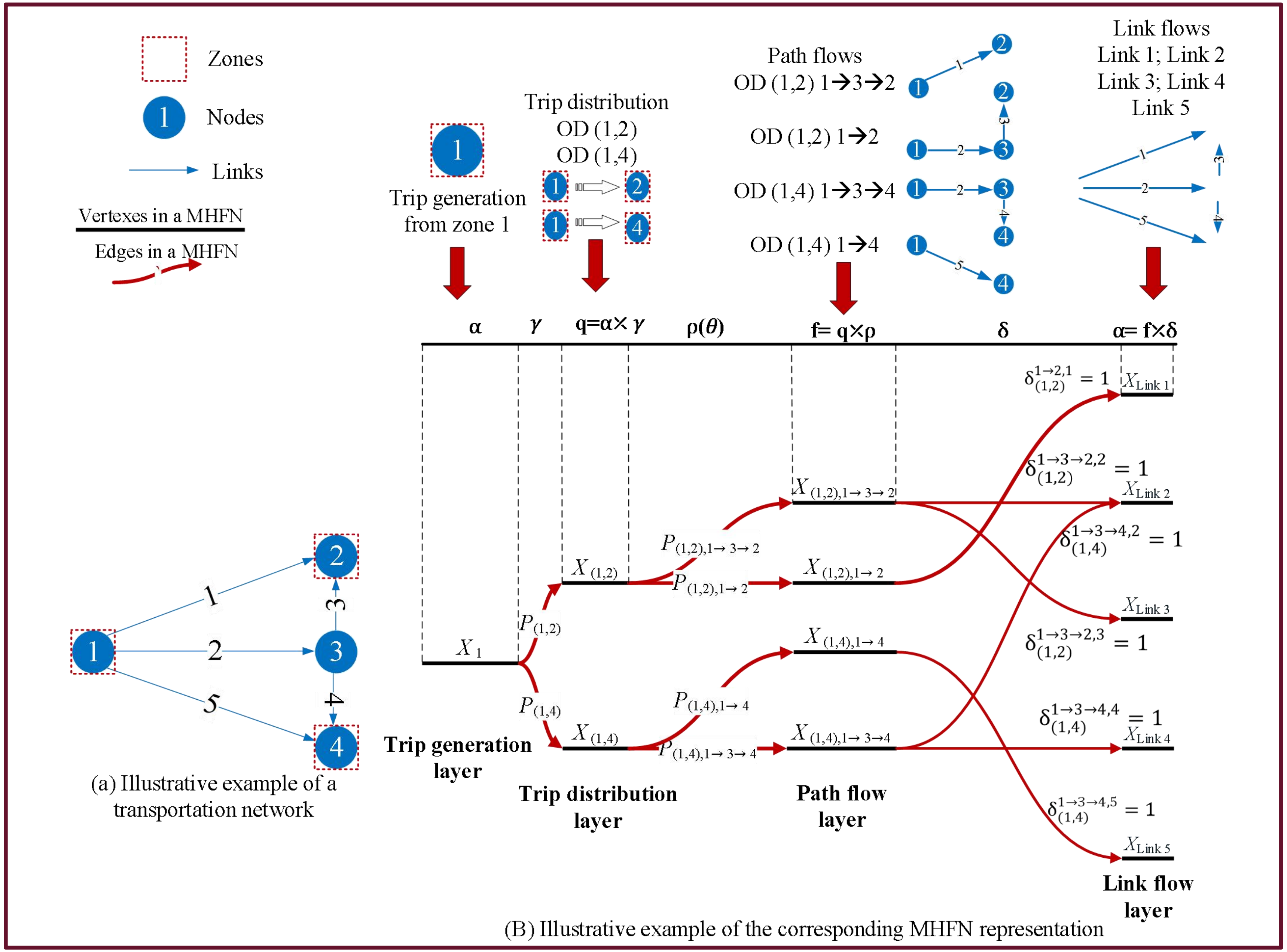


1. **The following flow conservation constraints are expressed by the "Neural Network" whose activation functions are ReLu function** $f(x) = \max(0, x)$:

$$X_o P_{ow} = X_w \;\; \forall\, w \in \mathcal{W} \;\; o \in \mathcal{Z}$$
$$X_w P_{wr} = X_r \;\; \forall\, r \in \mathcal{R} \;\; w \in \mathcal{W}$$
$$\sum_{r \in \mathcal{R}} \delta_{ra} X_r = X_a \;\; \forall a \in \mathcal{A}$$

2. Different types of data sources are mapped onto different layers of the architecture

Example of an HFN

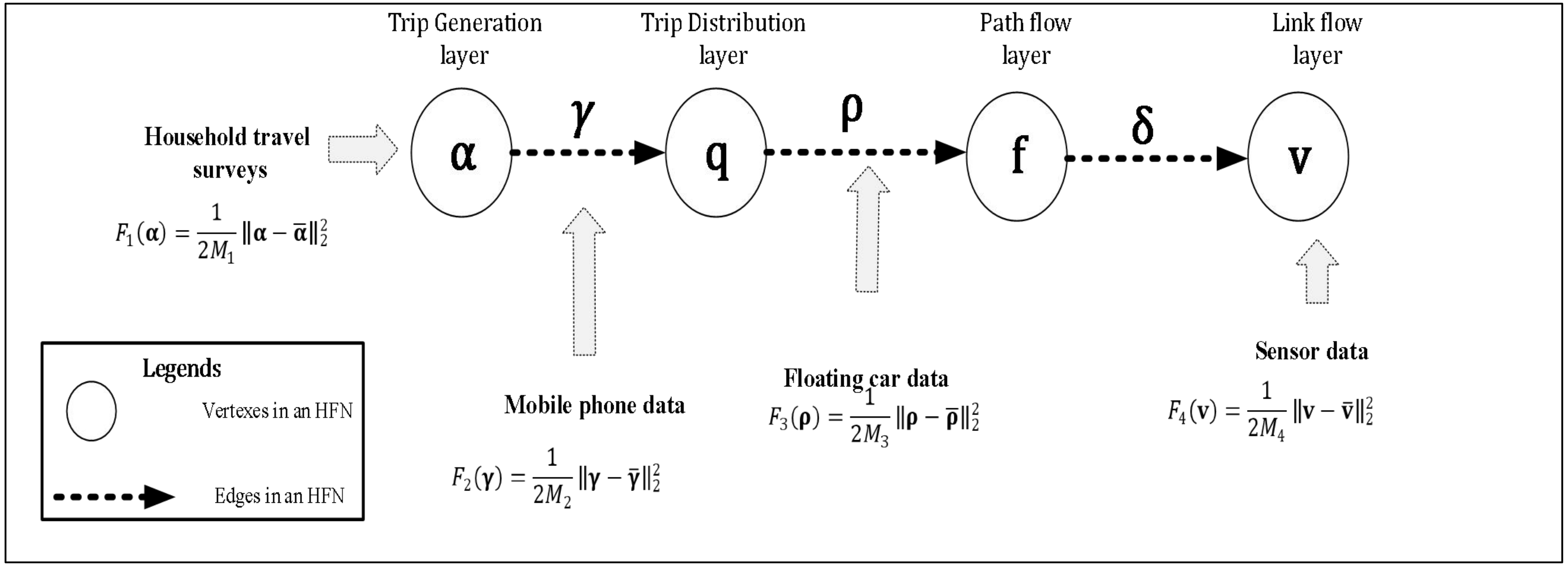


(a) Illustrative example of a transportation network

(B) Illustrative example of the corresponding MHFN representation

Arizona State University

HFN representation in a vectorized form



1. Different types of data sources generate different loss function:

$$F_1(\boldsymbol{\alpha}) = \frac{1}{2M_1}\|\boldsymbol{\alpha} - \bar{\boldsymbol{\alpha}}\|_2^2 \text{ (Survey)}$$

$$F_2(\boldsymbol{\gamma}) = \frac{1}{2M_2}\|\boldsymbol{\gamma} - \bar{\boldsymbol{\gamma}}\|_2^2 \text{ (Phone)}$$

$$F_3(\boldsymbol{\rho}) = \frac{1}{2M_3}\|\boldsymbol{\rho} - \bar{\boldsymbol{\rho}}\|_2^2 \text{ (GPS)}$$

$$F_4(\boldsymbol{v}) = \frac{1}{2M_4}\|\boldsymbol{v} - \bar{\boldsymbol{v}}\|_2^2 \text{ (Senor)}$$

$$\boldsymbol{\rho}_r = \frac{exp(U_r)}{\sum_{p\in \boldsymbol{P}(w)} exp(U_r)}$$

$$U_r = -\beta_{w,1}TC_r - \beta_{w,2}TT_r + \beta_w$$

Travel time of path r: TT_r

Travel cost of path r: TC_r

Multinomial logit model
(also known as **SOFTMAX function** in deep learning field)
In short
$\boldsymbol{\rho} = \text{softmax}(\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \boldsymbol{\beta}, \mathbf{TC}, \mathbf{TT})$

2. The 3 steps of the 4-step process can be expressed:

$$\boldsymbol{\alpha} \times \boldsymbol{\gamma} = \mathbf{q}$$

$$\mathbf{q} \times \boldsymbol{\rho} = \mathbf{f}$$

$$\mathbf{f} \times \boldsymbol{\delta} = \mathbf{v}$$

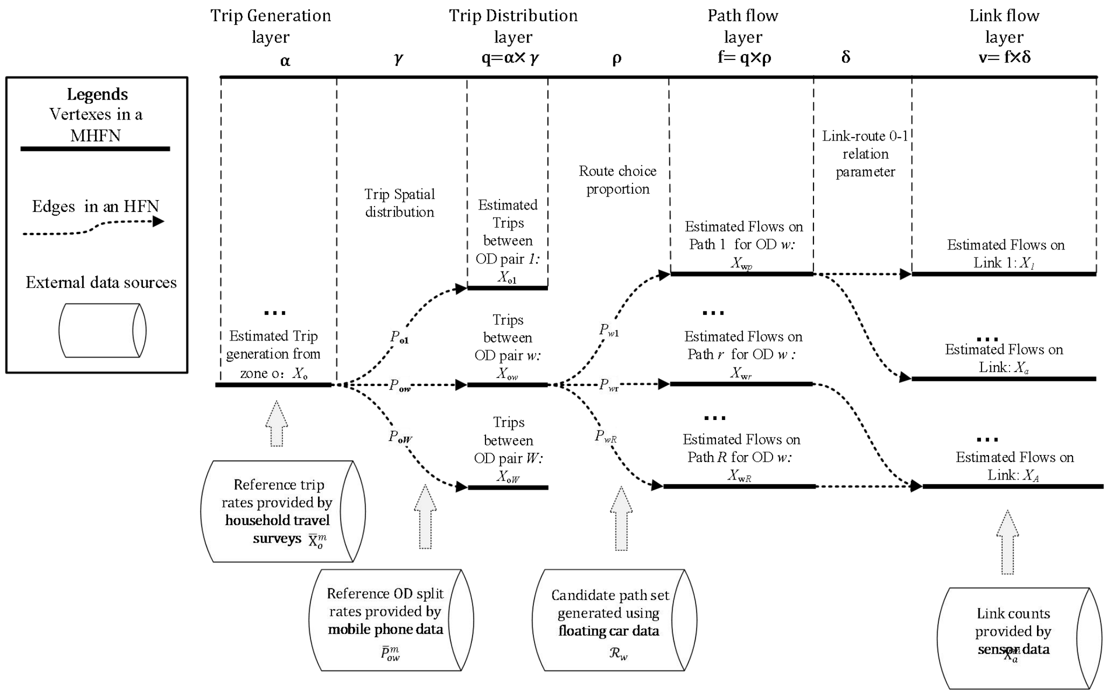
$$\boldsymbol{\rho} = \text{softmax}(\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \boldsymbol{\beta}, \mathbf{TC}, \mathbf{TT})$$

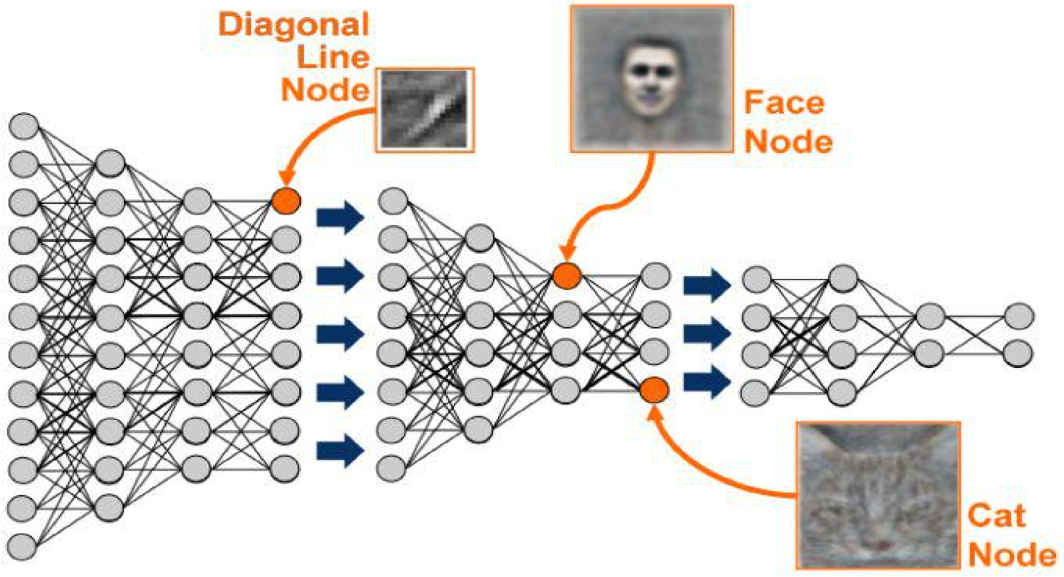
where $\boldsymbol{\alpha}$, $\mathbf{q}$, $\mathbf{f} \geq 0$; $\boldsymbol{\gamma}$, $\boldsymbol{\rho} \in [\mathbf{0}, \mathbf{1}]$

$\boldsymbol{\delta}$ is the path-link incident matrix

Arizona State University

Compare HFN with Artificial Neural Networks





Similarity:

- Hierarchy of the model
- Transitivity (Error signals are
- Forward and backward propagated on the network)

Differences:

- Data are input on different layers in HFNs
- Each "Neuron" has explainable traffic meaning

Arizona State University

3.2 Big data driven Transportation Computational Graph (BTCG)

Problem statement

Notations:

Traffic network

With hat → with observed traffic measurement

MHFN/ BTCG

o : origin
d: destination
r: route
a: link
m: sample
v : vertex in MHFN/ BTCG
e : edges in MHFN/ BTCG

Samples from Data sources

Sets	Definition
$\mathcal{N}$	Set of nodes in the transportation network
$\mathcal{A}$	Set of links in the transportation network
$\dot{\mathcal{A}}$	Subset of links with sensors in the transportation network
$\mathcal{Z}$	Set of zones that produces trips in the transportation network
$\dot{\mathcal{Z}}$	Subset of zones that have target trip generation generated from household travel survey.
$\mathcal{W}$	Set of OD pairs in the transportation network
$\mathcal{W}_o$	Set of OD pairs originating from $o \in \mathcal{Z}$ in the transportation network
$\bar{\mathcal{W}}$	Subset of OD pairs that have target OD split generated from mobile phone data
$\bar{\mathcal{W}}_o$	Subset of OD pairs starting from $o \in \mathcal{Z}$ that have target OD split generated from mobile phone data
$\mathcal{R}$	Set of routes in the transportation network
$\mathcal{R}_w$	Set of routes for OD pair $w = (o,d)$, where $\bigcup_{w\in\mathcal{W}} \mathcal{R}_w = \mathcal{R}$
V	Set of vertexes in the MHFN representation, $V = \mathcal{Z} \cup \mathcal{W} \cup \mathcal{R} \cup \mathcal{L}$
E	Set of edges in the MHFN representation
E_{ZW}	Set of edges in the MHFN representation relating zones and OD pairs
E_{WR}	Set of edges in the MHFN representation relating OD pairs and routes
E_{RA}	Set of edges in the MHFN representation relating routes and links
V_c	Set of vertexes in the computational graph corresponding a MHFN representation
E_c	Set of edges in the computational graph corresponding a MHFN representation
Indexes	**Definitions**
o	Indexes of production zones in Z
d	Indexes of destination zones in Z

Problem statement

Model formulation

$\mathcal{G} = (\mathcal{N}, \mathcal{A})$

The index of samples

Observed traffic measures

(1) **Target trip generation** $\overline{\mathrm{X}}_o^m$

(2) **Target OD split rate** $\overline{P}_{ow}^m$

(3) **Target flow counts** $\overline{\mathrm{X}}_a^m$

Demand variables in different levels

(1) X_o Estimated trip generations from zone $o \in \mathcal{Z}$ (car trip).

(2) X_w Estimated OD volumes of $w \in \mathcal{W}$ (car trip).

(3) X_r Estimated path flows loaded on path $r \in \mathcal{R}$ (car trip).

(4) X_a Estimated link flows on links $a \in \mathcal{A}$ (including links with and without sensors) (car trip).

Probability

(5) P_{ow} Estimated probability of users who passes OD $w \in \mathcal{W}$ from $o \in \mathcal{Z}$, $\sum_{w \in \mathcal{W}} P_{ow} = 1$. For $w \notin \overline{\mathcal{W}}_o$, $P_{ow} \equiv 0$.

(6) p_{ow} Estimated significance of OD w in set $\mathcal{W}_o$.

(7) P_{wr} Estimated probability of users who passes OD $w \in \mathcal{W}$ using $r \in \mathcal{R}_w$ $\sum_{r \in \mathcal{R}} P_{wr} = 1$.

Route choice Behavior parameters

(8) θ_w Estimated value of time (VOT) to translate time to monetary cost (ten dollar/hour).

Problem statement

$$\min F(\boldsymbol{\alpha}, \boldsymbol{\gamma}, \mathbf{v}) = \min F_1(\boldsymbol{\alpha}) + F_2(\boldsymbol{\gamma}) + F_3(\mathbf{v})$$

Demand variables

$$(1)\ \boldsymbol{\alpha} = (X_o | o \in \mathcal{Z})$$
$$(2)\ \mathbf{q} = (X_w | w \in \mathcal{W})$$
$$(3)\ \mathbf{f} = (X_r | r \in \mathcal{R})$$
$$(4)\ \mathbf{v} = (X_a | a \in \mathcal{A})$$

Probability

$$(5)\ \boldsymbol{\gamma} = (P_{ow} | o \in \mathcal{Z}, w \in \mathcal{W})$$
$$(6)\ \boldsymbol{\rho} = (P_{wr} | w \in \mathcal{W}, r \in \mathcal{R})$$

Behavior parameters

$$(7)\ \mathbf{p} = (p_{ow} | o \in \mathcal{Z}, w \in \mathcal{W})$$
$$(8)\ \boldsymbol{\theta} = (\theta_w | w \in \mathcal{W})$$

$$F_1(\boldsymbol{\alpha}) = \frac{1}{2} \sum_{m=1}^{M_1} \sum_{o \in \mathcal{Z}} (X_o - \bar{X}_o^m)^2$$

$$F_2(\boldsymbol{\gamma}) = \frac{1}{2} \sum_{m=1}^{M_2} \sum_{o \in \mathcal{Z}} \sum_{w \in \mathcal{W}_o} (P_{ow} - \bar{P}_{ow}^m)^2$$

$$F_3(\mathbf{v}) = \sum_{m=1}^{M_3} \sum_{a \in \mathcal{A}} \frac{1}{2} (X_a - \bar{X}_a^m)^2$$

Three objective function

1 Household travel survey

Population * trip rate

2 Mobile phone data

Target OD split

3 Sensor data

Problem statement

TDFE-UB Model

$$\min \mathrm{F}(\boldsymbol{\alpha},\boldsymbol{\gamma},\mathbf{v})$$

$$\mathrm{X}_o P_{ow} = X_w \ \ \forall\, w \in \mathcal{W}\ o\ \in \mathcal{Z}$$

$$P_{ow} = \begin{cases} \dfrac{p_{ow}}{\sum_{w\in\mathcal{W}_o} p_{ow}}, & w \in \mathcal{W}_o, o \in Z \\ 0 \quad , & w \notin \mathcal{W}_o, o \in Z \end{cases}$$

$$X_w P_{wr} = X_r\ \forall\, r \in \mathcal{R}\ w \in \mathcal{W}$$

$$\sum_{r\in\mathcal{R}} \delta_{ra} X_r = \mathrm{X}_a \ \ \forall a \in \mathcal{A}$$

$$P_{wr} = \begin{cases} \dfrac{\exp(-\theta_w \sum_{a\in\mathcal{A}} \delta_{ra} t_a - \sum_{a\in\mathcal{A}} \delta_{ra} c_a)}{\sum_{r\in\mathcal{R}_w} \exp(-\theta_w \sum_{a\in\mathcal{A}} \delta_{ra} t_a - \sum_{a\in\mathcal{A}} \delta_{ra} c_a)}, & r \in \mathcal{R}_w, w \in W \\ 0 \quad , & r \notin \mathcal{R}_w, w \in W \end{cases}$$

$$\mathbf{v} = \mathbf{v}(\mathbf{f}) = \mathbf{v}\big(\mathbf{f}(\mathbf{q},\boldsymbol{\rho})\big) = \mathbf{v}\Big(\mathbf{f}\big(\mathbf{q}(\boldsymbol{\alpha},\boldsymbol{\gamma}(\mathbf{p})),\boldsymbol{\rho}(\boldsymbol{\theta})\big)\Big)$$
$$= \boldsymbol{\alpha} \times \boldsymbol{\gamma}(\mathbf{p}) \times \boldsymbol{\rho}(\boldsymbol{\theta}) \times \boldsymbol{\delta}$$

$$\mathrm{X}_a = \sum_{o\in\mathcal{Z}} \sum_{w\in\mathcal{W}} \sum_{r\in\mathcal{R}} \mathrm{X}_o P_{ow} P_{wr} \delta_{ra} \ \ \forall a \in \mathcal{A}$$

- Composite function, comprising of Non linear function
- Non convex

$$\min \mathrm{F}(\boldsymbol{\alpha},\boldsymbol{\gamma},\mathbf{v})$$

$$\boldsymbol{\alpha} \times \boldsymbol{\gamma}(\mathbf{p}) = \mathbf{q}$$

$$\mathbf{q} \times \boldsymbol{\rho}(\boldsymbol{\theta}) = \mathbf{f}$$

$$\mathbf{f} \times \boldsymbol{\delta} = \mathbf{v}$$

$$\min \mathrm{F}(\boldsymbol{\alpha},\boldsymbol{\gamma},\mathbf{v}) = \min \mathrm{F}_1(\boldsymbol{\alpha}) + \mathrm{F}_2(\boldsymbol{\gamma})$$
$$+ \mathrm{F}_3\Big(\mathbf{v}\Big(\mathbf{f}\big(\mathbf{q}(\boldsymbol{\alpha},\boldsymbol{\gamma}(\mathbf{p})),\boldsymbol{\rho}(\boldsymbol{\theta})\big)\Big)\Big)$$

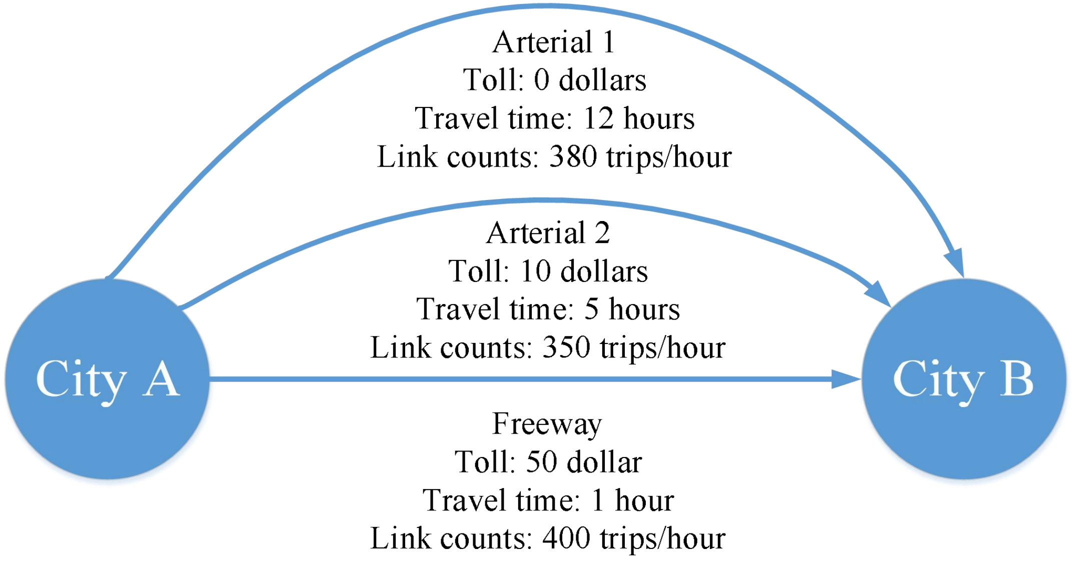
Problem statement
TDFE-UB Model

$$F_1(X_o,\theta)=\frac{1}{2}\left(X_{freeway}-400\right)^2+\frac{1}{2}\left(X_{arterial\ 1}-380\right)^2+\frac{1}{2}\left(X_{arterial\ 2}-350\right)^2$$

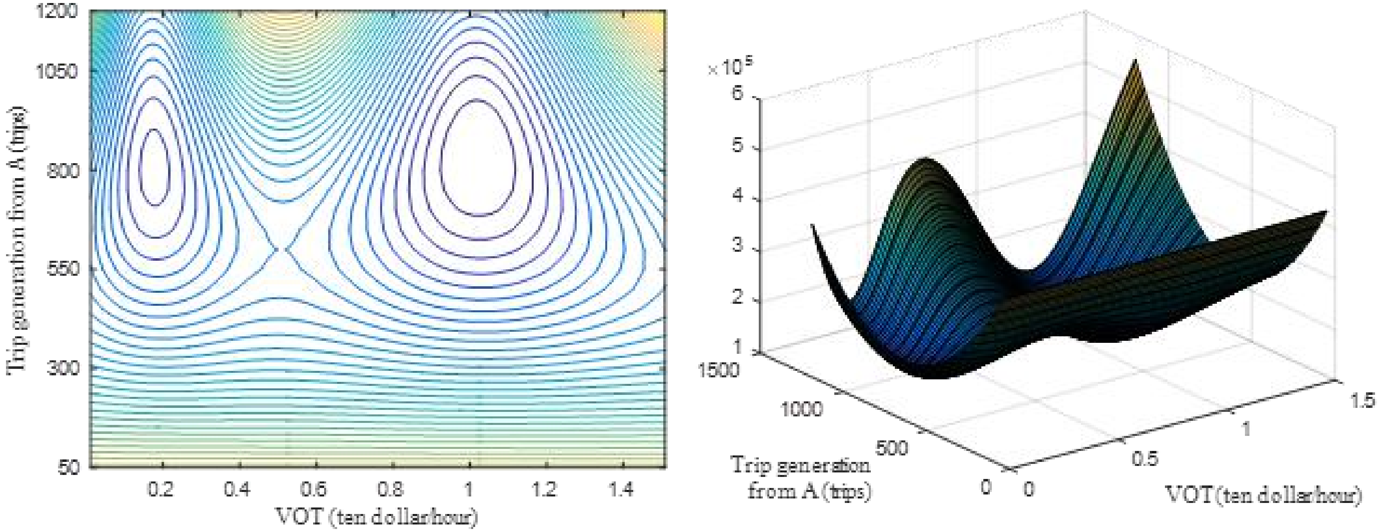
$$X_o\times\frac{\exp(-1\times\theta-5)}{\exp(-1\times\theta-5)+\exp(-5\times\theta-1)+\exp(-12\times\theta)}=X_{freeway}$$

$$X_o\times\frac{\exp(-5\times\theta-1)}{\exp(-1\times\theta-5)+\exp(-5\times\theta-1)+\exp(-12\times\theta)}=X_{arterial\ 1}$$

$$X_o\times\frac{\exp(-12\times\theta)}{\exp(-1\times\theta-5)+\exp(-5\times\theta-1)+\exp(-12\times\theta)}=X_{arterial\ 2}$$

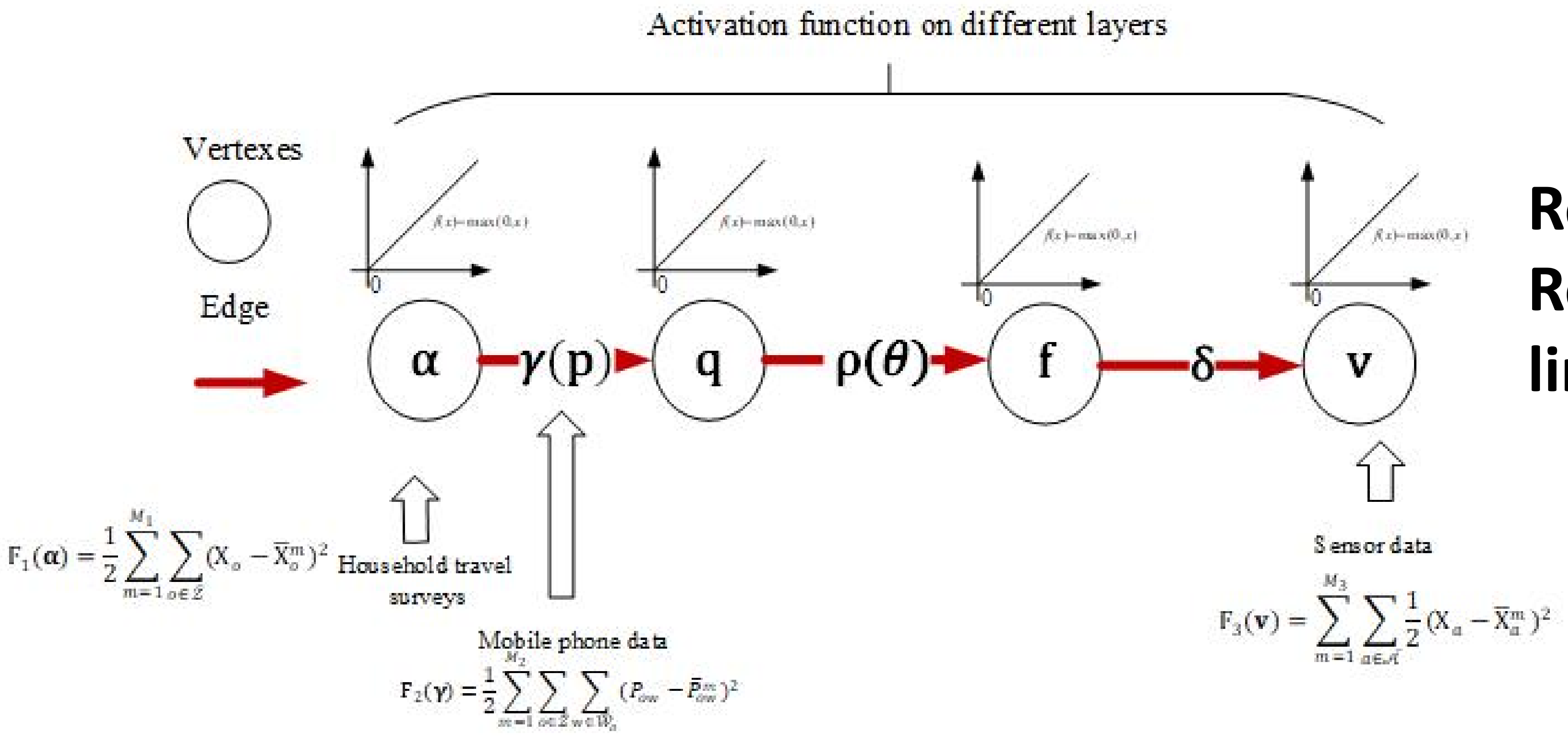


- Non-convex



(A) Contour line of the objective function (B) Values of the objective function versus trip generation and VOT

Reformulation using multi-layer hierarchical flow network representation



(a) Four layer MHFN is actually a neural network with ReLU activation function on different layers

ReLU
Rectified linear unit

Reformulation using multi-layer hierarchical flow network representation

universal approximation theorem

In the mathematical theory of artificial neural networks, the **universal approximation theorem** states[1] that a feed-forward network with a single hidden layer containing a finite number of neurons (i.e., a multilayer perceptron), can approximate continuous functions on compact subsets of $\mathbf{R}^n$, under mild assumptions on the activation function. T

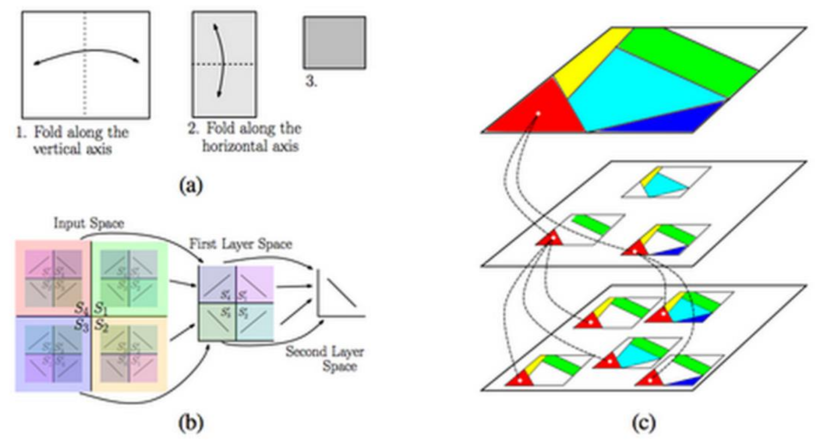


Figure 2: (a) Space folding of 2-D Euclidean space along the two coordinate axes. (b) An illustration of how the top-level partitioning (on the right) is replicated to the original input space (left). (c) Identification of regions across the layers of a deep model.

ReLU

Rectified linear unit: Why do not need Logistic sigmoid?

Logistic sigmoid

Drawbacks: saturate

ReLu

Advantage: Fast

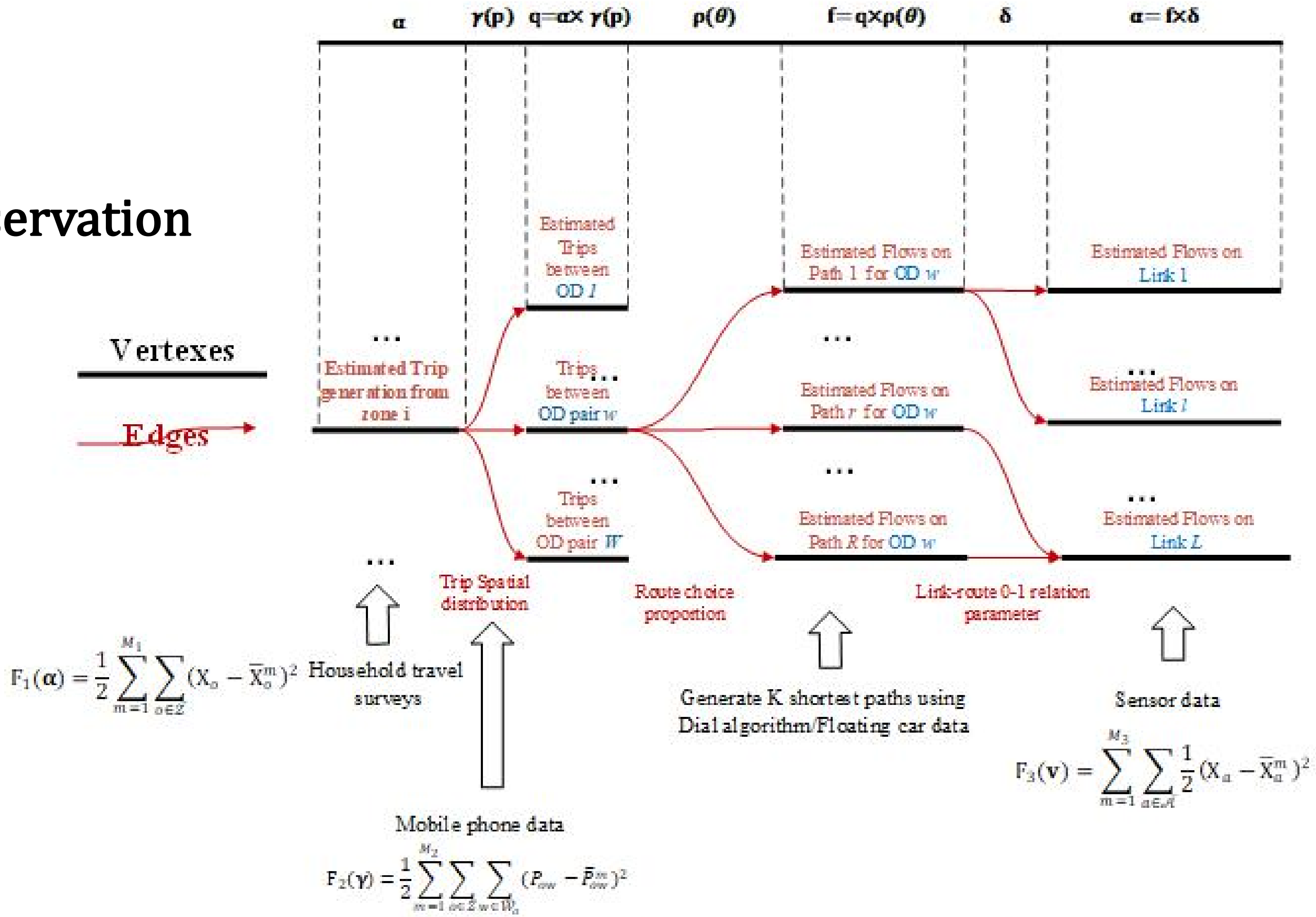
Reformulation using multi-layer hierarchical flow network representation

Multi-commodity flow conservation constraints

$$X_o = \sum_{w \in \mathcal{W}_o} X_w \;\; \forall\, o \in \mathcal{Z}$$

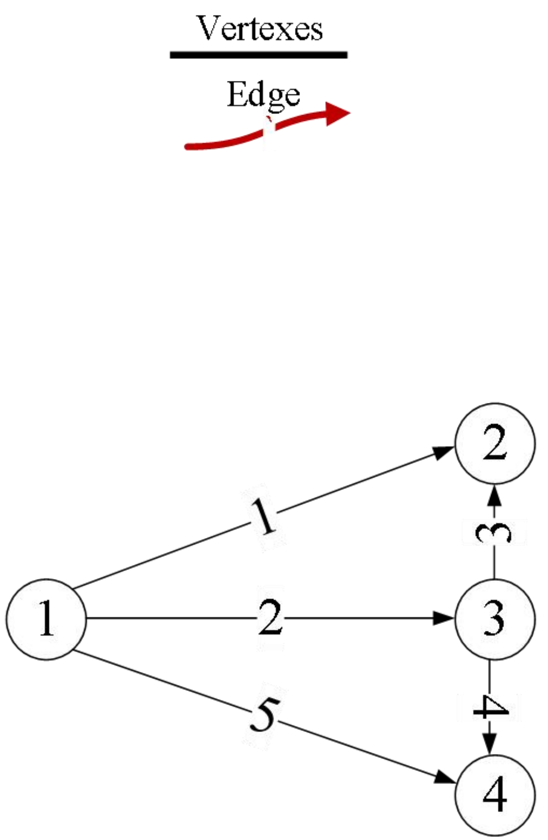
$$X_w = \sum_{r \in \mathcal{R}_w} X_r \;\; \forall\, w \in \mathcal{W}$$

$$\sum_{w \in \mathcal{W}} \sum_{r \in \mathcal{R}_w} \delta_{ra} \times X_r = X_a \;\; \forall a \in \mathcal{A}$$

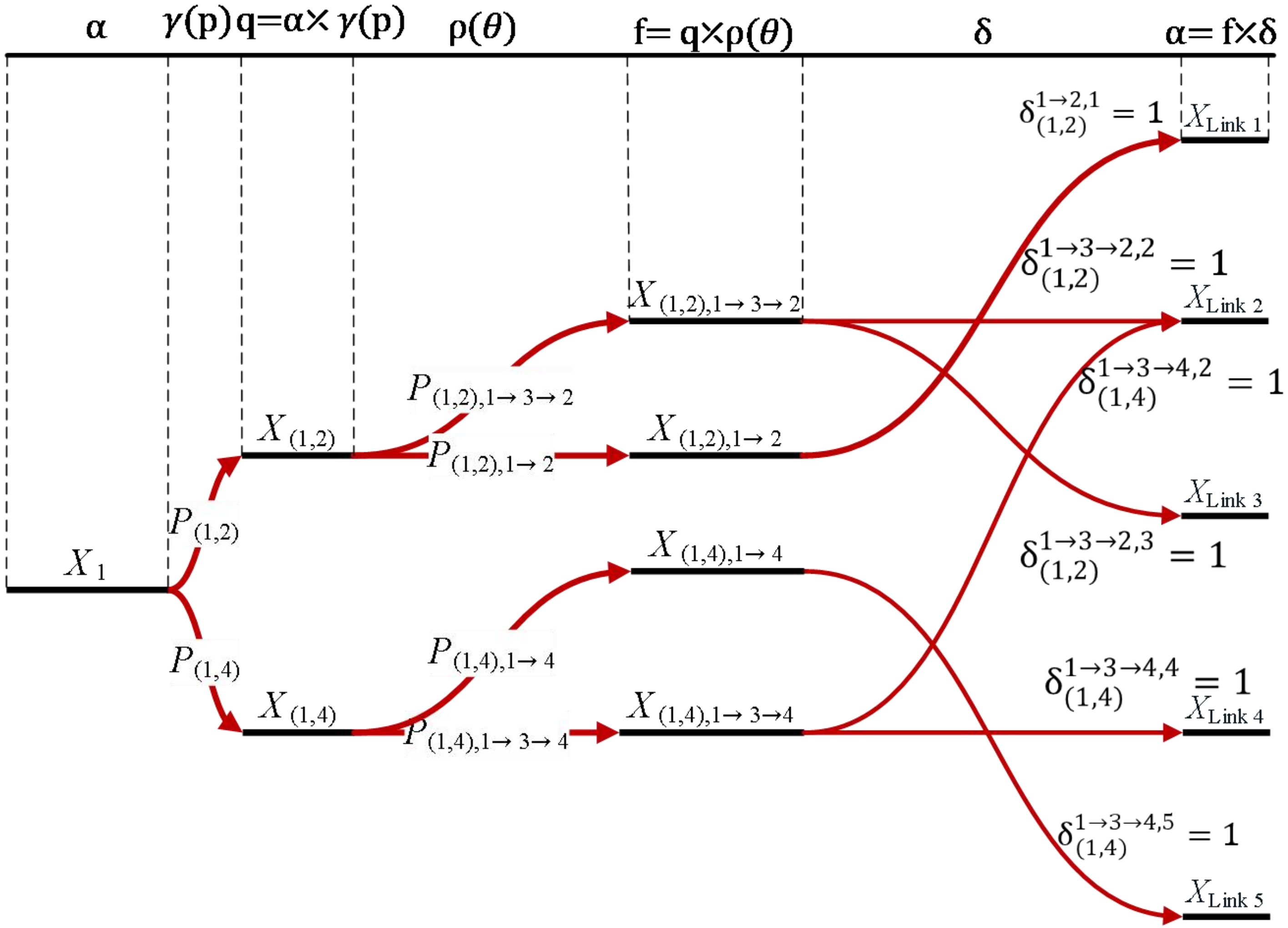


(b) The relationship between different data sources and the MHFN representation

Reformulation using multi-layer hierarchical flow network representation



(a) Illustrative example of a transportation network



(B) Illustrative example of the corresponding MHFN representation

Big data driven transportation computational graph to implement back propagation algorithm

(1) **Forward passing**

The forward passing process sequentially implements trip generation, trip distribution estimation, and route-based traffic assignment, which is an analogous process of the four-step approach in the field of traffic management.

(2) **Backward propagation**:

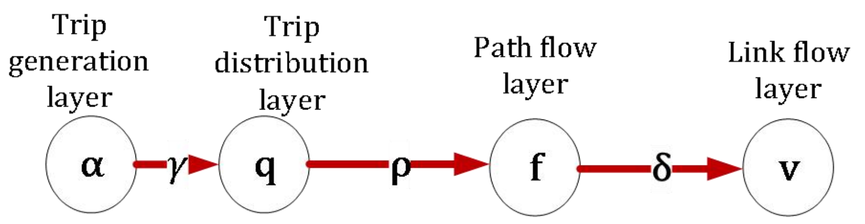
The back propagation process inversely implements a feedback control on the forward passing process. Different layers of first-order partial derivatives or “loss errors” are aggregated to calculate marginal gradients.

(3) **Variable values update**:

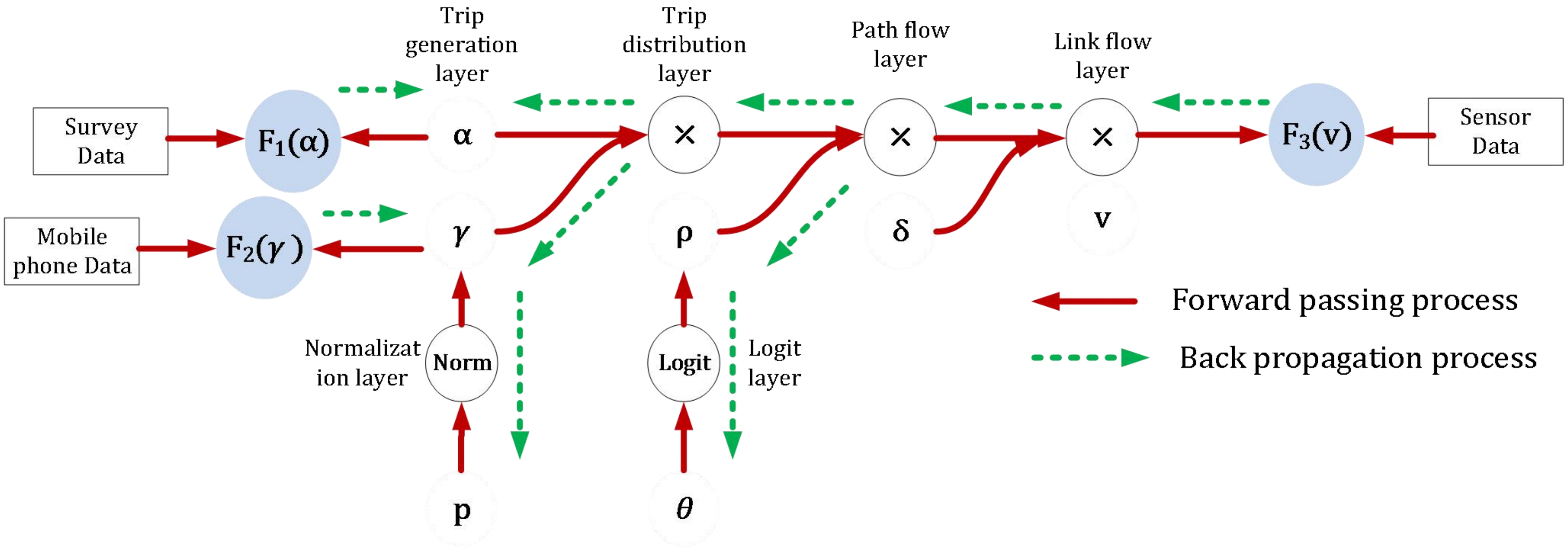
The estimation variables are updated using stochastic gradient descent using the marginal gradients.

Big data driven transportation computational graph to implement back propagation algorithm

Computational graph language to describe the back propagation



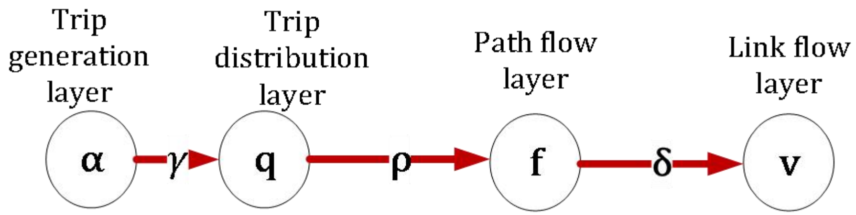
(a) MHFN of the TDFE-UB model



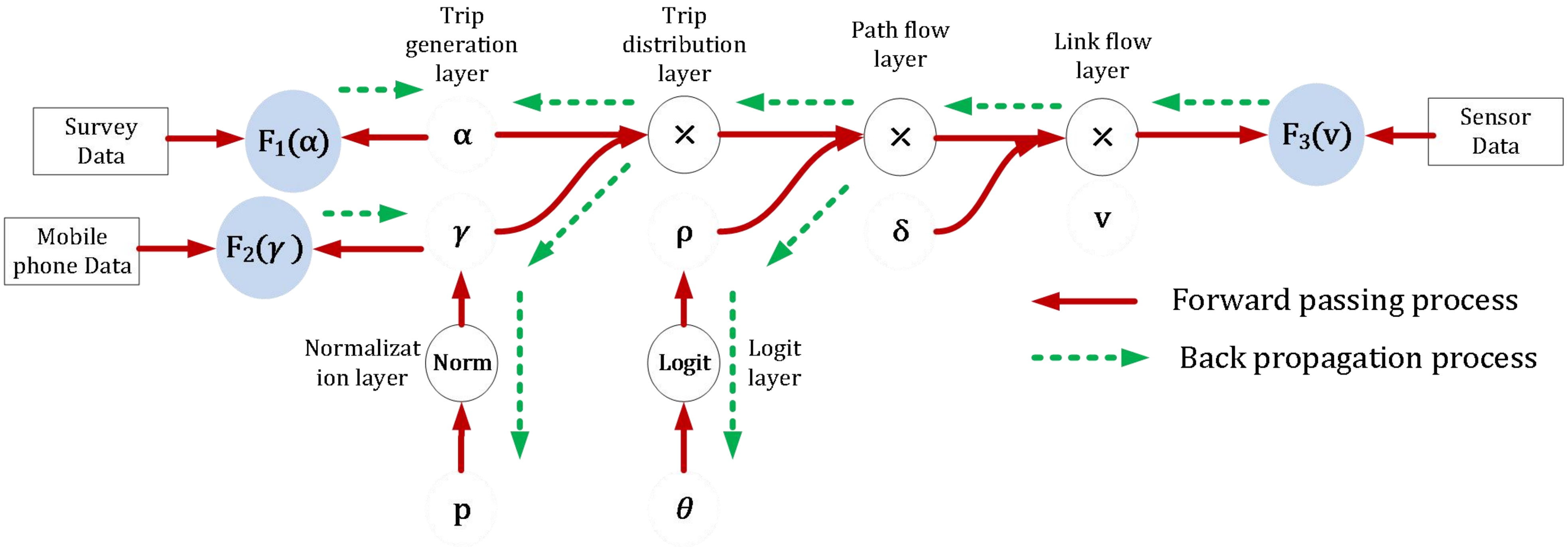
(b) The corresponding computational graph for the MHFN of the TDFE-UB model

Big data driven transportation computational graph to implement back propagation algorithm

Computational graph language to describe the back propagation



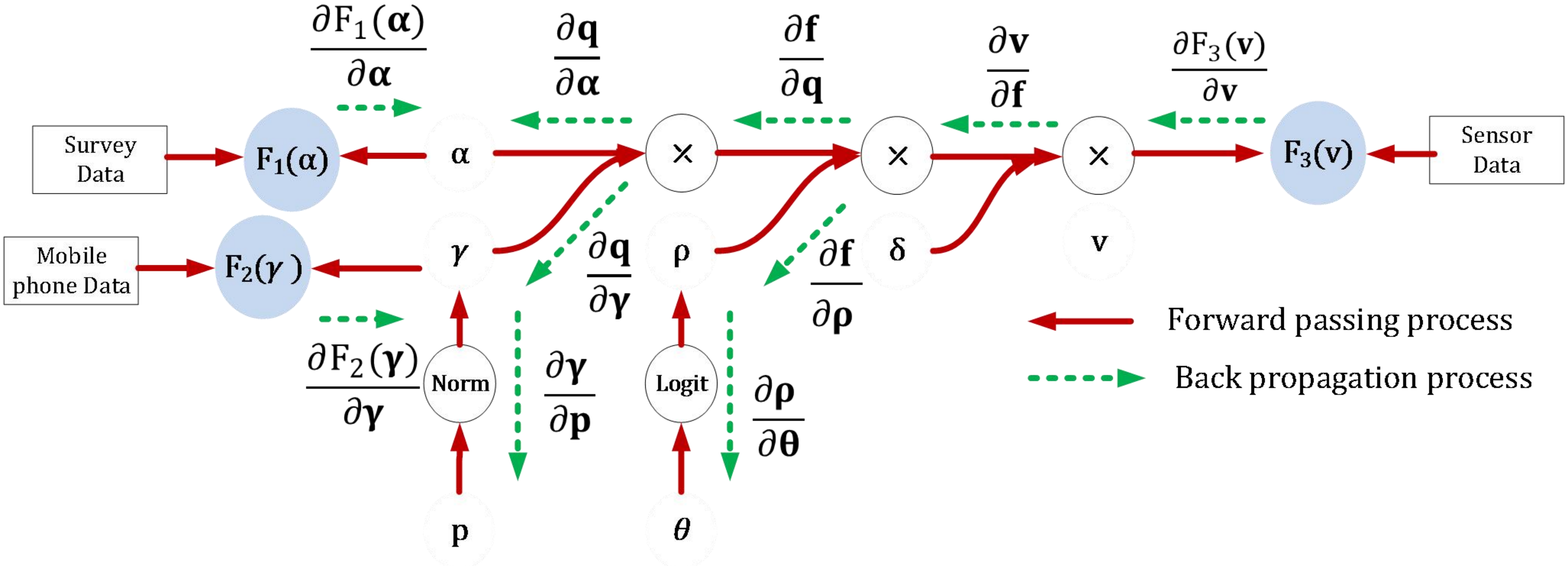
(a) MHFN of the TDFE-UB model



(b) The corresponding computational graph for the MHFN of the TDFE-UB model

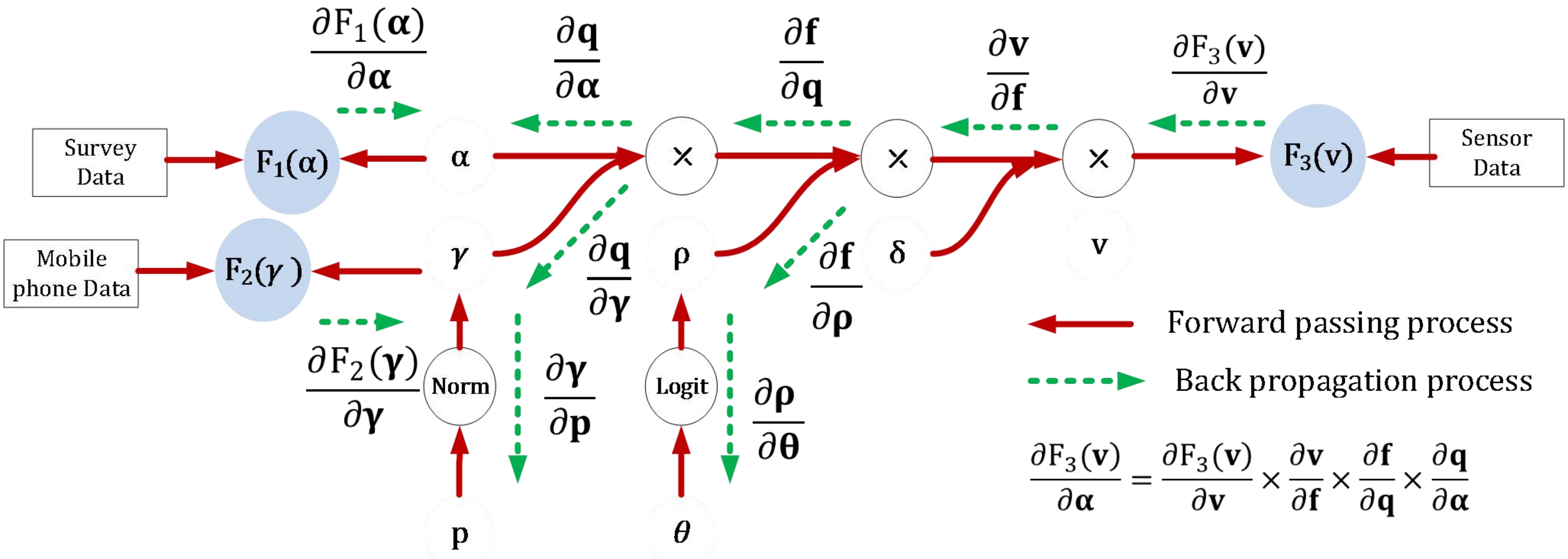
Big data driven transportation computational graph to implement back propagation algorithm

Computational graph language to describe the back propagation



Big data driven transportation computational graph to implement back propagation algorithm

Computational graph language to describe the back propagation



$$\frac{\partial F_3(v)}{\partial \alpha} = \frac{\partial F_3(v)}{\partial v} \times \frac{\partial v}{\partial f} \times \frac{\partial f}{\partial q} \times \frac{\partial q}{\partial \alpha}$$

$$\frac{\partial F_3(v)}{\partial p} = \frac{\partial F_3(v)}{\partial v} \times \frac{\partial v}{\partial f} \times \frac{\partial f}{\partial q} \times \frac{\partial q}{\partial \gamma} \times \frac{\partial \gamma}{\partial p}$$

$$\frac{\partial F_3(v)}{\partial \theta} = \frac{\partial F_3(v)}{\partial v} \times \frac{\partial v}{\partial f} \times \frac{\partial f}{\partial \rho} \times \frac{\partial \rho}{\partial \theta}$$

Big data driven transportation computational graph to implement back propagation algorithm

Computational graph language to describe the back propagation

(1) Partial derivatives of loss functions with respect to estimation variables

(i) $\frac{\partial \mathrm{F}_1(\boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}}$ is a $|\mathcal{Z}|$ dimension vectors of partial derivatives.

$$\frac{\partial \mathrm{F}_1(\boldsymbol{\alpha})}{\partial X_o} = \sum_{m=1}^{M_1} (\mathrm{X}_o - \bar{\mathrm{X}}_o^m) \ \forall o \in \bar{\mathcal{Z}}$$

$$\frac{\partial \mathrm{F}_1(\boldsymbol{\alpha})}{\partial X_o} \equiv 0 \text{ Otherwise}$$

(ii) $\frac{\partial \mathrm{F}_2(\boldsymbol{\gamma})}{\partial \boldsymbol{\gamma}}$ is a $|\mathcal{Z}| \times |\mathcal{W}|$ matrix of partial derivatives.

$$\frac{\partial \mathrm{F}_2(\boldsymbol{\gamma})}{\partial P_{ow}} = \sum_{m=1}^{M_2} (P_{ow} - \bar{P}_{ow}^m) \ \forall w \in \bar{\mathcal{W}}_o \ o \in \mathcal{Z}$$

$$\frac{\partial \mathrm{F}_2(\boldsymbol{\gamma})}{\partial P_{ow}} \equiv 0 \text{ Otherwise}$$

(iii) $\frac{\partial \mathrm{F}_3(\mathbf{v})}{\partial \mathbf{v}}$ is a $|\mathcal{A}|$ dimension vectors of partial derivatives, where

$$\frac{\partial \mathrm{F}_3(\mathbf{v})}{\partial X_a} = \sum_{m=1}^{M_3} (\mathrm{X}_a - \bar{\mathrm{X}}_a^m) \ \forall a \in \bar{\mathcal{A}}$$

$$\frac{\partial \mathrm{F}_3(\mathbf{v})}{\partial X_a} \equiv 0 \text{ Otherwise}$$

Big data driven transportation computational graph to implement back propagation algorithm

Computational graph language to describe the back propagation

(2) **Partial derivatives of a demand variable with respect to another demand variable**

(i) $\frac{\partial \mathbf{v}}{\partial \mathbf{f}}$ is a $|\mathcal{A}| \times |\mathcal{R}|$ Jacobian matrix, where the value of the a^{th} row and r^{th} column is

$$\frac{\partial X_a}{\partial X_r} = \delta_{ra} \; \forall a \in \mathcal{A}, r \in \mathcal{R}$$

(ii) $\frac{\partial \mathbf{f}}{\partial \mathbf{q}}$ is the $|\mathcal{R}| \times |\mathcal{W}|$ Jacobian matrix, where the value of the r^{th} row and w^{th} column is

$$\frac{\partial X_r}{\partial X_w} = P_{wr} \; \forall r \in \mathcal{R}, w \in \mathcal{W}$$

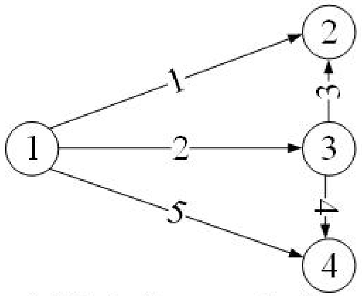
(iii) $\frac{\partial \mathbf{q}}{\partial \boldsymbol{\alpha}}$ is the $|\mathcal{W}| \times |\mathcal{A}|$ Jacobian matrix, where the value of the w^{th} row and o^{th} column is

$$\frac{\partial X_w}{\partial X_o} = P_{ow} \; \forall w \in \mathcal{W}, o \in \mathcal{Z}$$

Big data driven transportation computational graph to implement back propagation algorithm

Computational graph language to describe the back propagation

(3) **Partial derivatives of a demand variable with respect to a probability parameter**

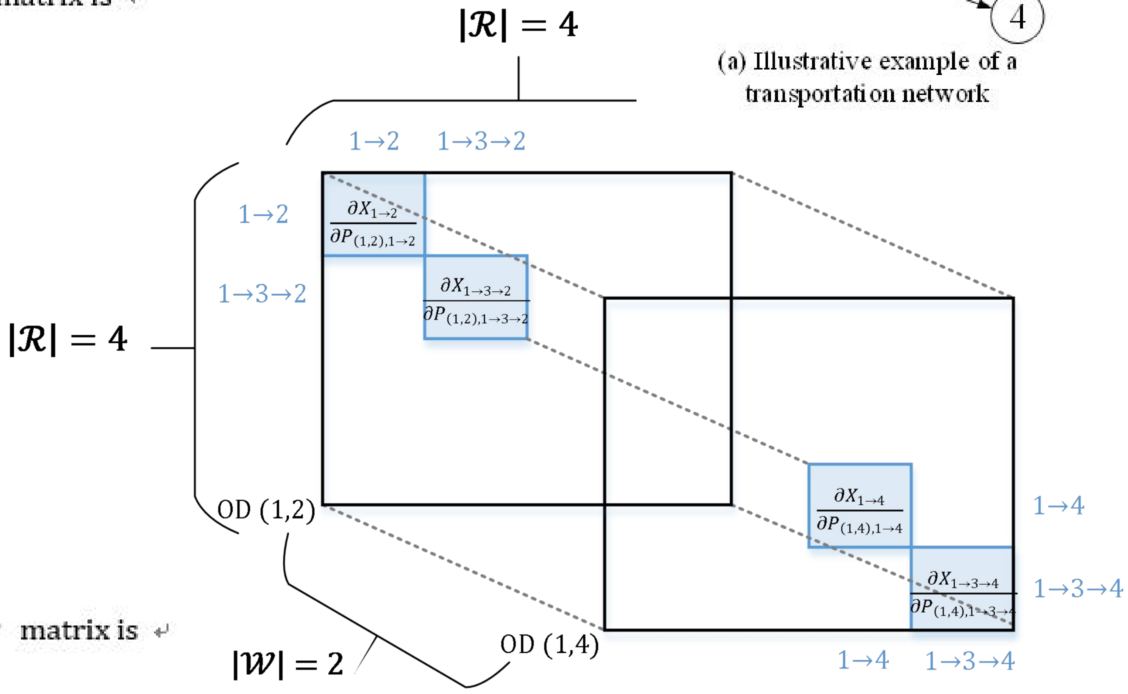


(a) Illustrative example of a transportation network

(i) $\frac{\partial f}{\partial \rho}$ is a $|\mathcal{R}| \times |\mathcal{W}| \times |\mathcal{R}|$ tensor, where the value of the r^{th} row and w^{th} column of the r'^{th} matrix is

$$\frac{\partial X_r}{\partial P_{wr'}} = X_w \;\forall\, r = r' \in \mathcal{R}_w, w \in \mathcal{W}$$

$$\frac{\partial X_r}{\partial P_{wr'}} \equiv 0 \text{ Otherwise}$$



(ii) $\frac{\partial q}{\partial \gamma}$ is a $|\mathcal{W}| \times |\mathcal{Z}| \times |\mathcal{W}|$ tensor, where the value of the w^{th} row and o^{th} column of the w'^{th} matrix is

$$\frac{\partial X_w}{\partial P_{ow'}} = X_o \;\forall\, w = w' \in \mathcal{W}_o, o \in \mathcal{Z}$$

$$\frac{\partial X_w}{\partial P_{ow'}} \equiv 0 \text{ Otherwise}$$

Big data driven transportation computational graph to implement back propagation algorithm

Computational graph language to describe the back propagation

(4) **Partial derivatives of a probability parameter with respect to a behavior parameter**

(i) $\frac{\partial \mathbf{p}}{\partial \boldsymbol{\theta}}$ is a $|\mathcal{R}| \times |\mathcal{W}|$ Jacobian matrix, where the value of the r^{th} row and w^{th} column is

$$\frac{\partial P_{wr}}{\partial \theta_w} = \frac{\exp\left(-\theta_w \sum_{a\in\mathcal{A}} \delta_{ra} t_a - \sum_{a\in\mathcal{A}} \delta_{ra} c_a\right) \sum_{r\in\mathcal{R}_w} [(\sum_{a\in\mathcal{A}} \delta_{ra} t_a) \exp(-\theta_w \sum_{a\in\mathcal{A}} \delta_{ra} t_a - \sum_{a\in\mathcal{A}} \delta_{ra} c_a)]}{[\sum_{r\in\mathcal{R}_w} \exp(-\theta_w \sum_{a\in\mathcal{A}} \delta_{ra} t_a - \sum_{a\in\mathcal{A}} \delta_{ra} c_a)]^2}$$

$$- \frac{(\sum_{a\in\mathcal{A}} \delta_{ra} t_a) \exp(-\theta_w \sum_{a\in\mathcal{A}} \delta_{ra} t_a - \sum_{a\in\mathcal{A}} \delta_{ra} c_a)}{\sum_{r\in\mathcal{R}_w} \exp(-\theta_w \sum_{a\in\mathcal{A}} \delta_{ra} t_a - \sum_{a\in\mathcal{A}} \delta_{ra} c_a)} \quad \forall r \in \mathcal{R}_w\ \forall w \in \mathcal{W};$$

$$\frac{\partial P_{wr}}{\partial \theta_w} \equiv 0 \text{ Otherwise}$$

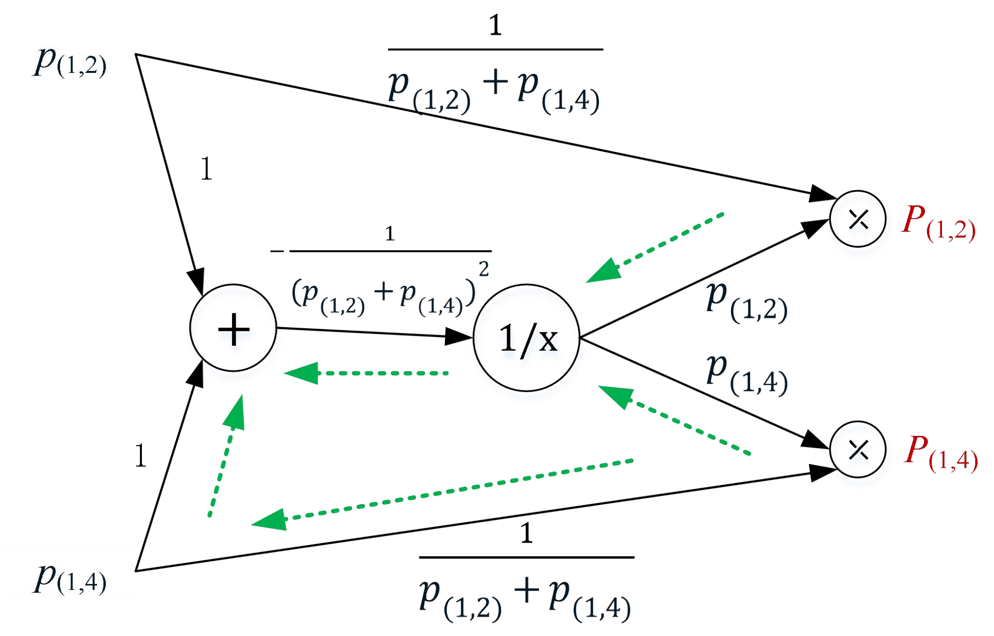
(ii) $\frac{\partial \boldsymbol{\gamma}}{\partial \mathbf{p}}$ is a $|\mathcal{W}| \times |\mathcal{W}|$ diagonal matrix, where the value of the w^{th} row and w'^{th} column is

$$\frac{\partial P_{ow}}{\partial p_{ow'}} = \frac{1}{\sum_{w\in\mathcal{W}_o} p_{ow}} - \frac{p_{ow}}{\left(\sum_{w\in\mathcal{W}_o} p_{ow}\right)^2} \quad \forall w = w' \in \mathcal{W}_o\ \forall o \in \mathcal{Z}$$

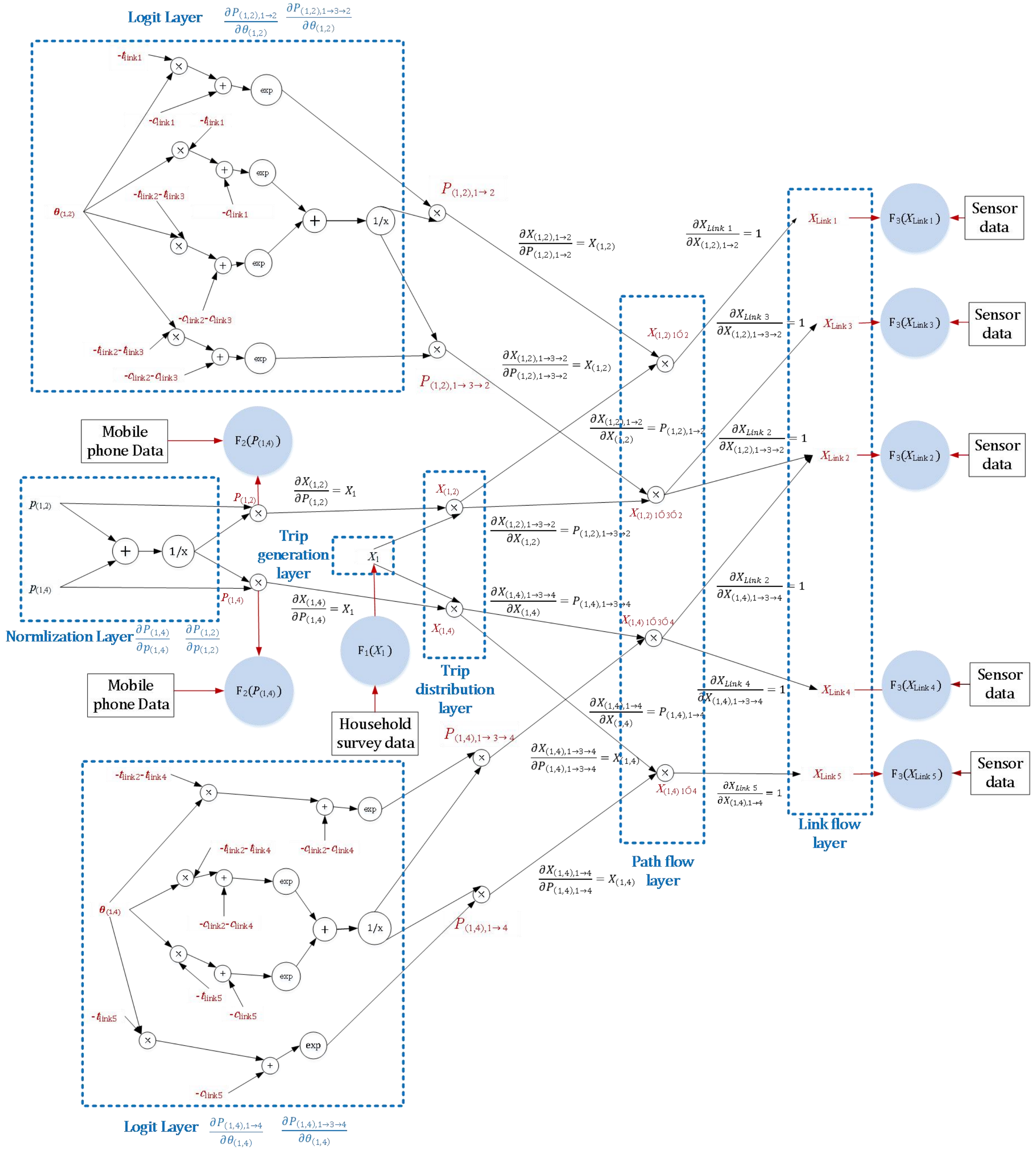
$$\frac{\partial P_{ow}}{\partial p_{ow'}} = -\frac{p_{ow}}{\left(\sum_{w\in\mathcal{W}_o} p_{ow}\right)^2} \quad \forall w \neq w' \in \mathcal{W}_o\ \forall o \in \mathcal{Z}$$

$$\frac{\partial P_{ow}}{\partial p_{ow}} \equiv 0 \text{ Otherwise}$$

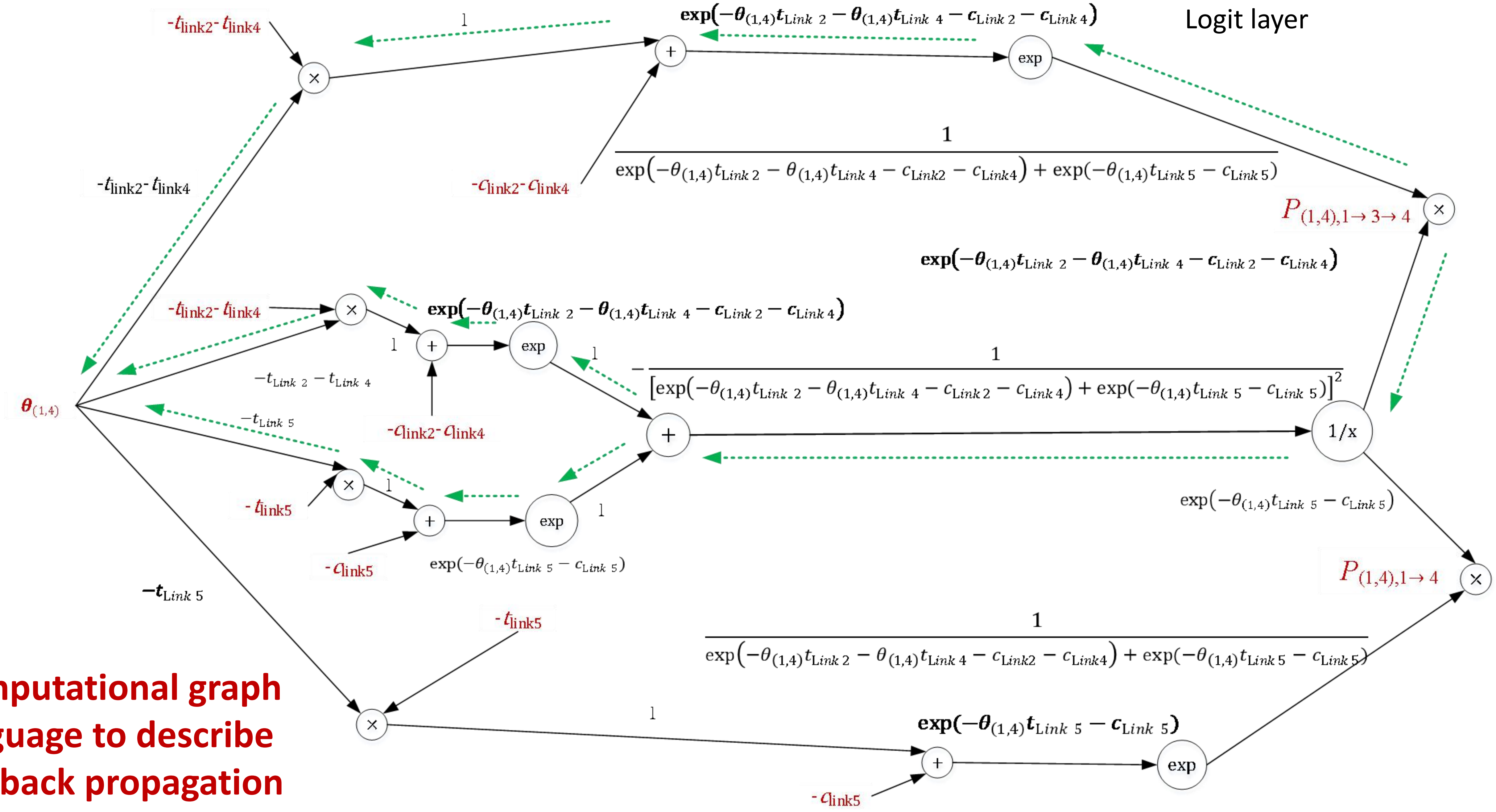
Computational graph language to describe the back propagation



Normalization layer



Logit layer



Computational graph language to describe the back propagation

Big data driven transportation computational graph to implement back propagation algorithm

Solution procedure to implement back propagation： exponent path from an input to an output

Dynamic programming

Bound condition: interface to data sources

$$\frac{\partial \mathrm{F}}{\partial v} = 1, \forall v \in \boldsymbol{V}_c^B$$

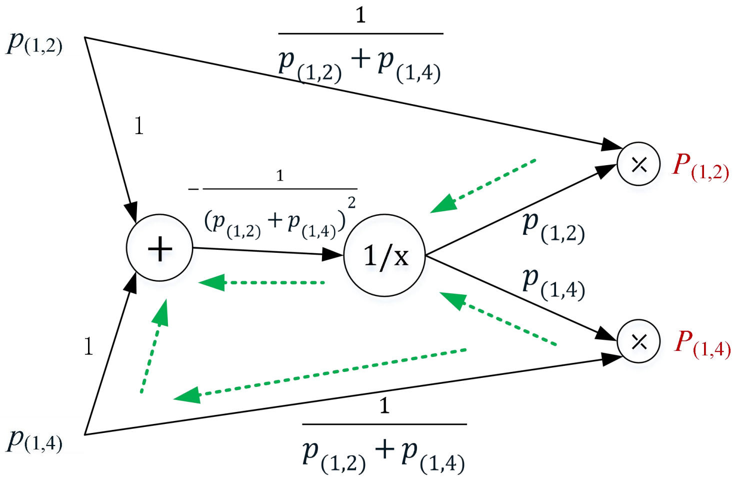
State transition function

$$\frac{\partial \mathrm{F}}{\partial v} = \sum_{v\prime \in \triangle^{+}(v)} \frac{\partial \mathrm{F}}{\partial v'} \frac{\partial v'}{\partial v}$$

Comparison

$F_j = \min_{i\prime \in \triangle^{-}(j)} (t_{ij} + F_i)$ shortest path

Let $\boldsymbol{V}_c$ denote the set of vertexes in the computational graph corresponding to MHFN $G(\boldsymbol{V}, \boldsymbol{A})$ (or the common term **“state”** in dynamic programming). $\boldsymbol{A}_c$ implies the set of edges in the computational graph corresponding to MHFN Bound conditions is set on the vertexes served as interface between data sources and the computational graph $G_c(\boldsymbol{V}_c, \boldsymbol{A}_c)$. Let $\boldsymbol{V}_c^B \subset \boldsymbol{V}_c$ be the set of bound vertexes.



$$\frac{\partial P_{(1,4)}}{\partial p_{(1,4)}} + \frac{\partial P_{(1,2)}}{\partial p_{(1,4)}} = \frac{1}{p_{(1,4)} + p_{(1,2)}} - \frac{p_{(1,4)}}{\left(p_{(1,4)} + p_{(1,2)}\right)^2} - \frac{p_{(1,2)}}{\left(p_{(1,4)} + p_{(1,2)}\right)^2} = 0$$

■ Backpropagation Algorithm

- **History**
- Backpropagation is a method to calculate the gradient of the loss function with respect to the weights in an artificial neural network.
- Backpropagation were derived in the context of control theory by Herry K. Kelley in 1960 and by Arthur E. Bryson in 1961, using principles of dynamic programming.
- **Intuition**
- Learning as an optimization problem.
- **Limitations**
- Gradient descent with backpropagation is not guaranteed to find the global minimum of the error function.

Big data driven transportation computational graph to implement back propagation algorithm

Solution procedure to implement back propagation

Dynamic programming

Bound condition: interface to data sources

$$\frac{\partial \mathrm{F}}{\partial v} = 1, \forall v \in \boldsymbol{V}_c^B$$

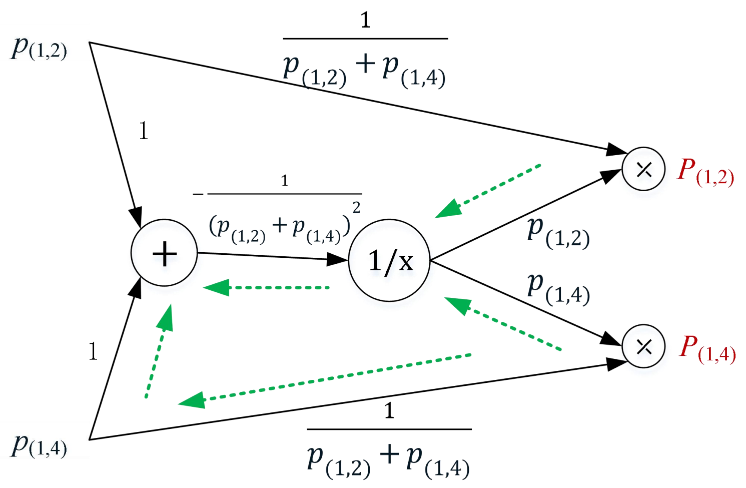
State transition function

$$\frac{\partial \mathrm{F}}{\partial v} = \sum_{v\prime \in \triangle^{+}(v)} \frac{\partial \mathrm{F}}{\partial v'} \frac{\partial v'}{\partial v}$$

Comparison

$F_j = \min_{i\prime \in \triangle^{-}(j)}(t_{ij} + F_i)$ shortest path

Let $\boldsymbol{V}_c$ denote the set of vertexes in the computational graph corresponding to MHFN $G(\boldsymbol{V}, \boldsymbol{A})$ (or the common term **"state"** in dynamic programming). $\boldsymbol{A}_c$ implies the set of edges in the computational graph corresponding to MHFN Bound conditions is set on the vertexes served as interface between data sources and the computational graph $G_c(\boldsymbol{V}_c, \boldsymbol{A}_c)$. Let $\boldsymbol{V}_c^B \subset \boldsymbol{V}_c$ be the set of bound vertexes.



$$\frac{\partial P_{(1,4)}}{\partial p_{(1,4)}} + \frac{\partial P_{(1,2)}}{\partial p_{(1,4)}} = \frac{1}{p_{(1,4)} + p_{(1,2)}} - \frac{p_{(1,4)}}{(p_{(1,4)} + p_{(1,2)})^2} - \frac{p_{(1,2)}}{(p_{(1,4)} + p_{(1,2)})^2} = 0$$

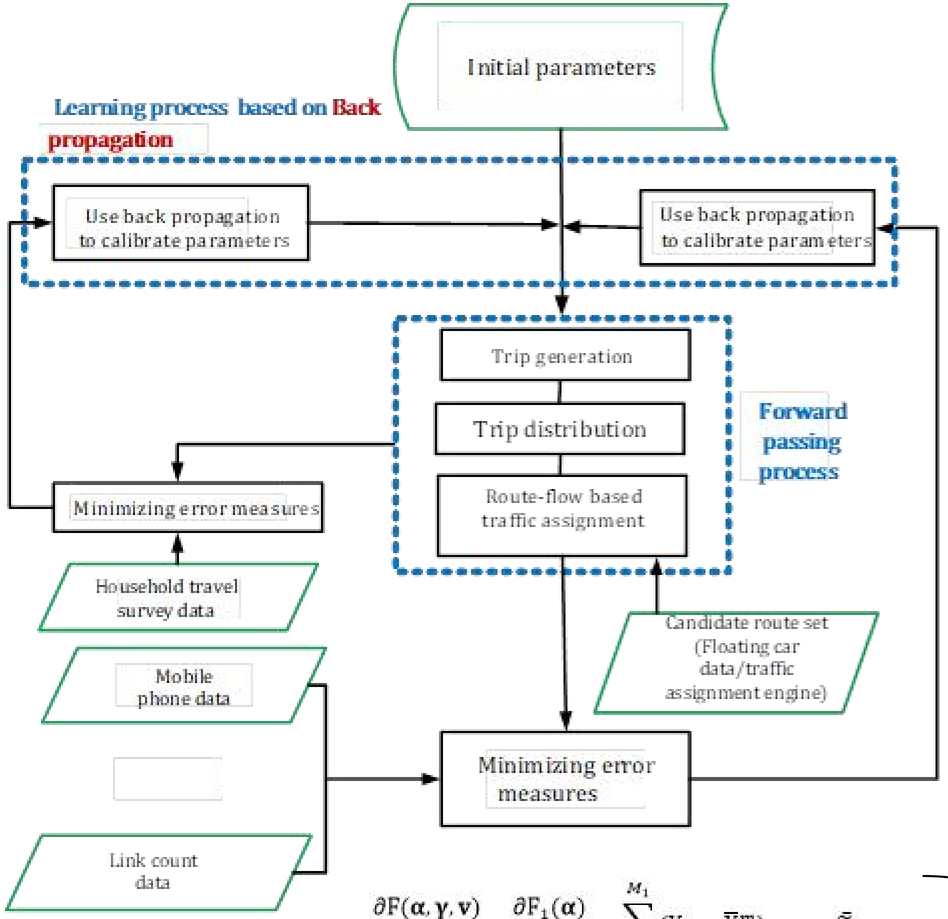
Backpropagation Algorithm

- Initialize weights (typically random!)
- Keep doing epochs
  - For each example **e** in training set do
    - **forward pass** to compute
      - O = neural-net-output(network,e)
      - miss = (T-O) at each output unit
    - **backward pass** to calculate deltas to weights
    - update all weights
  - end
- until tuning set error stops improving

Chain rule of derivatives

$$\frac{\partial E}{\partial w_{k,j}^{(2.1)}} = \frac{\partial E}{\partial o_k} \frac{\partial o_k}{\partial net_k^{(2)}} \frac{\partial net_k^{(2)}}{\partial w_{k,j}^{(2.1)}}$$

Solution procedure to implement back propagation



$$\frac{\partial F(\boldsymbol{\alpha},\boldsymbol{\gamma},\mathbf{v})}{\partial X_o}=\frac{\partial F_1(\boldsymbol{\alpha})}{\partial X_o}=\sum_{m=1}^{M_1}(X_o-\bar{X}_o^m)\ \forall o\in\bar{\mathcal{Z}}$$

$$\frac{\partial F(\boldsymbol{\alpha},\boldsymbol{\gamma},\mathbf{v})}{\partial P_{ow}}=\frac{\partial F_2(\boldsymbol{\gamma})}{\partial P_{ow}}=\sum_{m=1}^{M_2}(P_{ow}-\bar{P}_{ow}^m)\ \forall w\in\bar{\mathcal{W}}_o\ o\in\mathcal{Z}$$

$$\frac{\partial F(\boldsymbol{\alpha},\boldsymbol{\gamma},\mathbf{v})}{\partial X_a}=\frac{\partial F_3(\mathbf{v})}{\partial X_a}=\sum_{m=1}^{M_3}(X_a-\bar{X}_a^m)\ \forall a\in\bar{\mathcal{A}}$$

Bound conditions

Normalization

Step 1. **Initialize**

Step 2. **For epoch n do**

Step 2.1. Do forward passing step

Step 2.2. Backward propagate the deviation

For all o in $\mathcal{Z}$

For all w in $\mathcal{W}_o$

For all r in $\mathcal{R}_w$

$$\frac{\partial F_3(\mathbf{v})}{\partial X_r}=\sum_{a\in\bar{\mathcal{A}}}\frac{\partial F_3(\mathbf{v})}{\partial X_a}\times\frac{\partial X_a}{\partial X_r}=\sum_{a\in\bar{\mathcal{A}}}\delta_{ra}\times\frac{\partial F_3(\mathbf{v})}{\partial X_a}$$

$$\frac{\partial F_3(\mathbf{v})}{\partial P_{wr}}=\frac{\partial F_3(\mathbf{v})}{\partial X_r}\times\frac{\partial X_r}{\partial P_{wr}}=\frac{\partial F_3(\mathbf{v})}{\partial X_r}\times X_w=\frac{\partial F_3(\mathbf{v})}{\partial X_r}\times X_o\times P_{ow}$$

End

$$\frac{\partial F_3(\mathbf{v})}{\partial X_w}=\sum_{r\in\mathcal{R}_w}\frac{\partial F_3(\mathbf{v})}{\partial X_r}\times\frac{\partial X_r}{\partial X_w}=\sum_{r\in\mathcal{R}_w}\frac{\partial F_3(\mathbf{v})}{\partial X_r}\times P_{wr}$$

$$\frac{\partial F_3(\mathbf{v})}{\partial \theta_w}=\frac{\partial F_3(\mathbf{v})}{\partial X_r}\times\frac{\partial X_r}{\partial P_{wr}}\times\sum_{r\in\mathcal{R}_w}\frac{\partial P_{wr}}{\partial \theta_w}=\frac{\partial F_3(\mathbf{v})}{\partial X_r}\times X_o\times P_{ow}\times\sum_{r\in\mathcal{R}_w}\frac{\partial P_{wr}}{\partial \theta_w}$$

$$\frac{\partial F_3(\mathbf{v})}{\partial P_{ow}}=\frac{\partial F_3(\mathbf{v})}{\partial X_w}\times\frac{\partial X_w}{\partial P_{ow}}=\frac{\partial F_3(\mathbf{v})}{\partial X_w}\times X_o$$

$$\frac{\partial F_3(\mathbf{v})}{\partial p_{ow}}=\sum_{w'\in\mathcal{W}_o}\frac{\partial F_3(\mathbf{v})}{\partial X_w}\times\frac{\partial X_w}{\partial P_{ow}}\times\frac{\partial P_{ow'}}{\partial p_{ow}}=\sum_{w'\in\mathcal{W}_o}\frac{\partial F_3(\mathbf{v})}{\partial X_w}\times X_o\times\frac{\partial P_{ow'}}{\partial p_{ow}}\frac{\partial F_3(\mathbf{v})}{\partial X_w}$$

End

$$\frac{\partial F_3(\mathbf{v})}{\partial X_o}=\sum_{w\in\mathcal{W}_o}\frac{\partial F_3(\mathbf{v})}{\partial X_w}\times\frac{\partial X_w}{\partial X_o}=\sum_{w\in\mathcal{W}_o}\frac{\partial F_3(\mathbf{v})}{\partial X_w}\times P_{ow}$$

End

Step 2.3. Variable values update.

$$X_w=\max\left(0,X_w-\varphi_1\times\frac{\partial F_3(\mathbf{v})}{\partial P_{ow}}-\varphi_1\times\frac{\partial F_1(\boldsymbol{\alpha})}{\partial X_w}\right)\ \forall w\in\mathcal{W}$$

$$P_{ow}=\max\left(0,P_{ow}-\varphi_2\times\frac{\partial F_3(\mathbf{v})}{\partial P_{ow}}-\varphi_2\times\frac{\partial F_2(\boldsymbol{\gamma})}{\partial P_{ow}}\right)\ \forall w\in\mathcal{W}_o\ o\in\mathcal{Z}$$

$$\theta_w=\max\left(0,\ \theta_w-\varphi_3\times\frac{\partial F_3(\mathbf{v})}{\partial \theta_w}\right)\ \forall w\in\mathcal{W}$$

End

Step 3. **Stop until condition of convergence is satisfied.**

Euler algorithm
Stochastic Gradient descendent

Solution procedure to implement back propagation

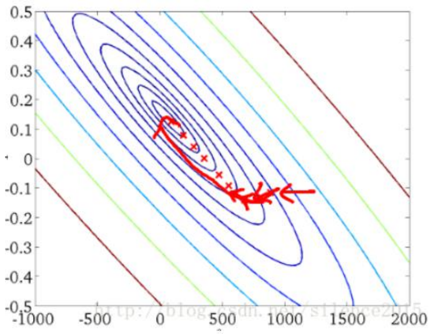
Batch Gradient Descent

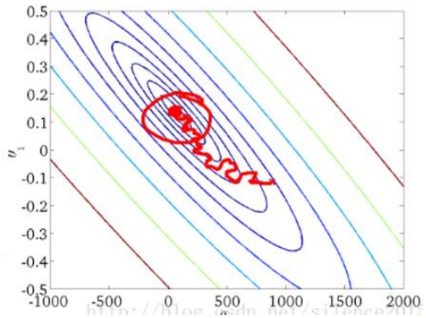
repeat until convergence: {

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_j^{(i)} \qquad \text{for } j := 0..n$$

}

Stochastic Gradient Descent





Mini-batch Gradient Descent

Repeat:

For $i = 1, 11, 21, 31, \ldots, 991$

$$\theta_j := \theta_j - \alpha \frac{1}{10} \sum_{k=i}^{i+9} (h_\theta(x^{(k)}) - y^{(k)}) x_j^{(k)}$$

Step 1. Initialize

Step 2. For epoch n do

Step 2.1. Do forward passing step

Step 2.2. Backward propagate the deviation

For all o in $\mathcal{Z}$

For all w in $\mathcal{W}_o$

For all r in $\mathcal{R}_w$

$$\frac{\partial F_3(\mathbf{v})}{\partial X_r} = \sum_{a \in \bar{\mathcal{A}}} \frac{\partial F_3(\mathbf{v})}{\partial X_a} \times \frac{\partial X_a}{\partial X_r} = \sum_{a \in \bar{\mathcal{A}}} \delta_{ra} \times \frac{\partial F_3(\mathbf{v})}{\partial X_a}$$

$$\frac{\partial F_3(\mathbf{v})}{\partial P_{wr}} = \frac{\partial F_3(\mathbf{v})}{\partial X_r} \times \frac{\partial X_r}{\partial P_{wr}} = \frac{\partial F_3(\mathbf{v})}{\partial X_r} \times X_w = \frac{\partial F_3(\mathbf{v})}{\partial X_r} \times X_o \times P_{ow}$$

End

$$\frac{\partial F_3(\mathbf{v})}{\partial X_w} = \sum_{r \in \mathcal{R}_w} \frac{\partial F_3(\mathbf{v})}{\partial X_r} \times \frac{\partial X_r}{\partial X_w} = \sum_{r \in \mathcal{R}_w} \frac{\partial F_3(\mathbf{v})}{\partial X_r} \times P_{wr}$$

$$\frac{\partial F_3(\mathbf{v})}{\partial \theta_w} = \frac{\partial F_3(\mathbf{v})}{\partial X_r} \times \frac{\partial X_r}{\partial P_{wr}} \times \sum_{r \in \mathcal{R}_w} \frac{\partial P_{wr}}{\partial \theta_w} = \frac{\partial F_3(\mathbf{v})}{\partial X_r} \times X_o \times P_{ow} \times \sum_{r \in \mathcal{R}_w} \frac{\partial P_{wr}}{\partial \theta_w}$$

$$\frac{\partial F_3(\mathbf{v})}{\partial P_{ow}} = \frac{\partial F_3(\mathbf{v})}{\partial X_w} \times \frac{\partial X_w}{\partial P_{ow}} = \frac{\partial F_3(\mathbf{v})}{\partial X_w} \times X_o$$

$$\frac{\partial F_3(\mathbf{v})}{\partial p_{ow}} = \sum_{w' \in \mathcal{W}_o} \frac{\partial F_3(\mathbf{v})}{\partial X_w} \times \frac{\partial X_w}{\partial P_{ow}} \times \frac{\partial P_{ow'}}{\partial p_{ow}} = \sum_{w' \in \mathcal{W}_o} \frac{\partial F_3(\mathbf{v})}{\partial X_w} \times X_o \times \frac{\partial P_{ow'}}{\partial p_{ow}} \frac{\partial F_3(\mathbf{v})}{\partial X_w}$$

End

$$\frac{\partial F_3(\mathbf{v})}{\partial X_o} = \sum_{w \in \mathcal{W}_o} \frac{\partial F_3(\mathbf{v})}{\partial X_w} \times \frac{\partial X_w}{\partial X_o} = \sum_{w \in \mathcal{W}_o} \frac{\partial F_3(\mathbf{v})}{\partial X_w} \times P_{ow}$$

End

Step 2.3. Variable values update.

$$X_w = \max(0, X_w - \varphi_1 \times \frac{\partial F_3(\mathbf{v})}{\partial P_{ow}} - \varphi_1 \times \frac{\partial F_1(\boldsymbol{\alpha})}{\partial X_w}) \ \forall w \in \mathcal{W}$$

$$P_{ow} = \max(0, P_{ow} - \varphi_2 \times \frac{\partial F_3(\mathbf{v})}{\partial P_{ow}} - \varphi_2 \times \frac{\partial F_2(\boldsymbol{\gamma})}{\partial P_{ow}}) \ \forall w \in \mathcal{W}_o\ o \in \mathcal{Z}$$

$$\theta_w = \max(0,\ \theta_w - \varphi_3 \times \frac{\partial F_3(\mathbf{v})}{\partial \theta_w}) \ \forall w \in \mathcal{W}$$

End

Step 3. Stop until condition of convergence is satisfied.

Euler algorithm
Stochastic Gradient descendent

Solution procedure to implement back propagation

(1) Vanishing gradient problem in BTCG

$$\frac{\partial F_3(\mathbf{v})}{\partial \boldsymbol{\alpha}} = \frac{\partial F_3(\mathbf{v})}{\partial \mathbf{v}} \times \frac{\partial \mathbf{v}}{\partial \mathbf{f}} \times \frac{\partial \mathbf{f}}{\partial \mathbf{q}} \times \frac{\partial \mathbf{q}}{\partial \boldsymbol{\alpha}}$$

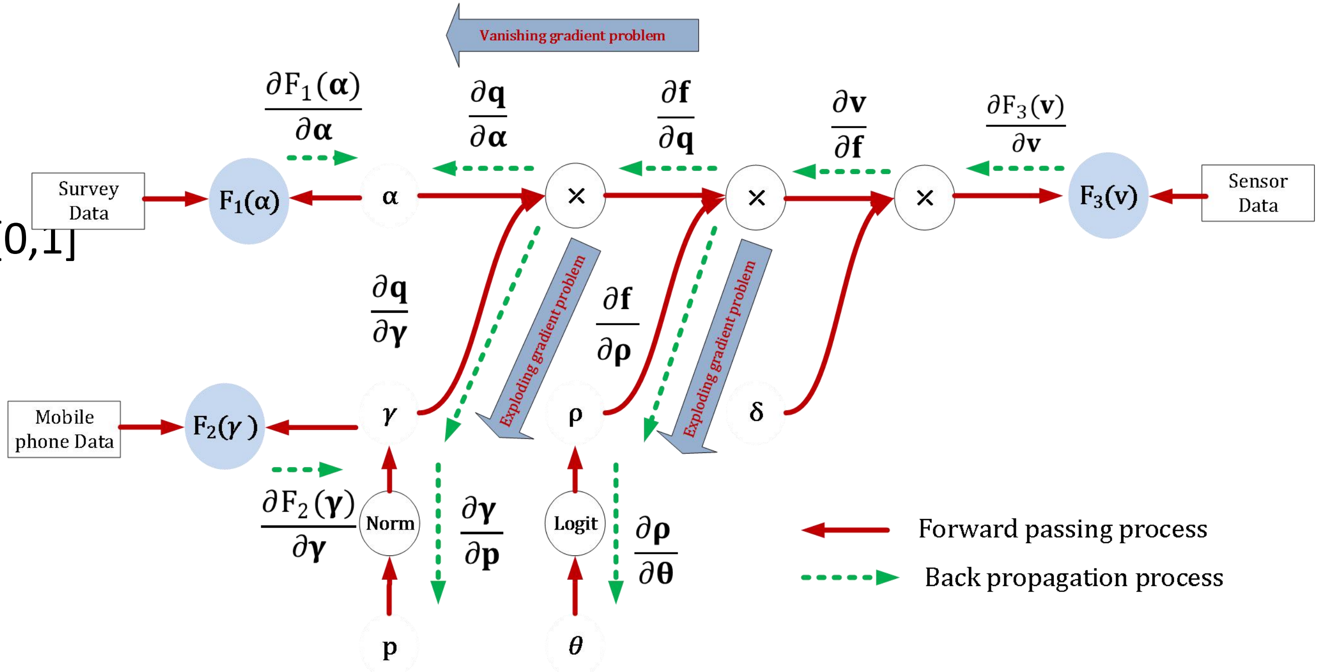
Small values within range [0,1]

(2) Exploding gradient problem in BTCG

$$\frac{\partial F_3(\mathbf{v})}{\partial \mathbf{p}} = \frac{\partial F_3(\mathbf{v})}{\partial \mathbf{v}} \times \frac{\partial \mathbf{v}}{\partial \mathbf{f}} \times \frac{\partial \mathbf{f}}{\partial \mathbf{q}} \times \frac{\partial \mathbf{q}}{\partial \boldsymbol{\gamma}} \times \frac{\partial \boldsymbol{\gamma}}{\partial \mathbf{p}}$$

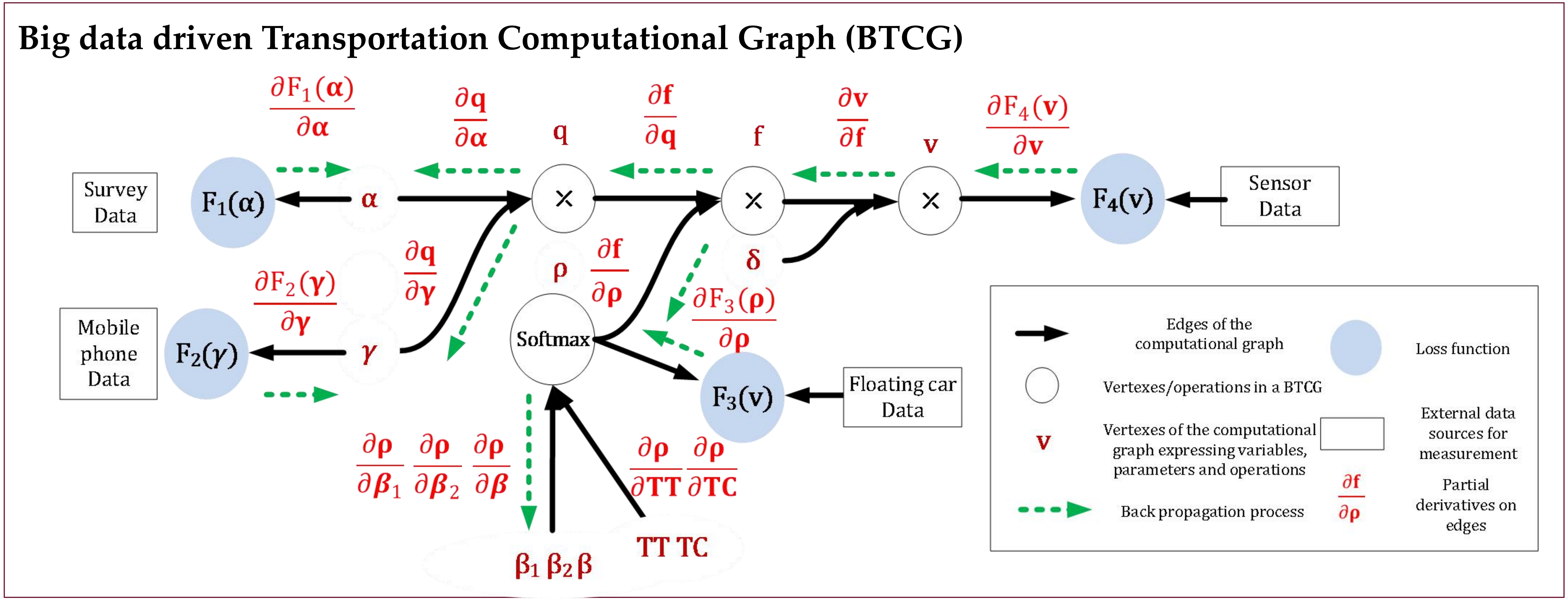
$$\frac{\partial F_3(\mathbf{v})}{\partial \boldsymbol{\theta}} = \frac{\partial F_3(\mathbf{v})}{\partial \mathbf{v}} \times \frac{\partial \mathbf{v}}{\partial \mathbf{f}} \times \frac{\partial \mathbf{f}}{\partial \boldsymbol{\rho}} \times \frac{\partial \boldsymbol{\rho}}{\partial \boldsymbol{\theta}}$$

Large values equal to OD volume or trip production



$\varphi_3 \gg \varphi_2 > \varphi_1$

Extend an HFN to a computational graph



Q1: How sensor data impact behavioral coefficients?

$$\frac{\partial F_4(\mathbf{v})}{\partial \beta} = \frac{\partial F_4(\mathbf{v})}{\partial \mathbf{v}} \frac{\partial \mathbf{v}}{\partial \mathbf{f}} \frac{\partial \mathbf{f}}{\partial \rho} \frac{\partial \rho}{\partial \beta}$$

Q2: How parameter γ is calibrated by different two kinds of traffic measurements?

$$(1)\ \frac{\partial F_4(\mathbf{v})}{\partial \gamma} = \frac{\partial F_4(\mathbf{v})}{\partial \mathbf{v}} \frac{\partial \mathbf{v}}{\partial \mathbf{f}} \frac{\partial \mathbf{f}}{\partial \mathbf{q}} \frac{\partial \mathbf{q}}{\partial \gamma}$$

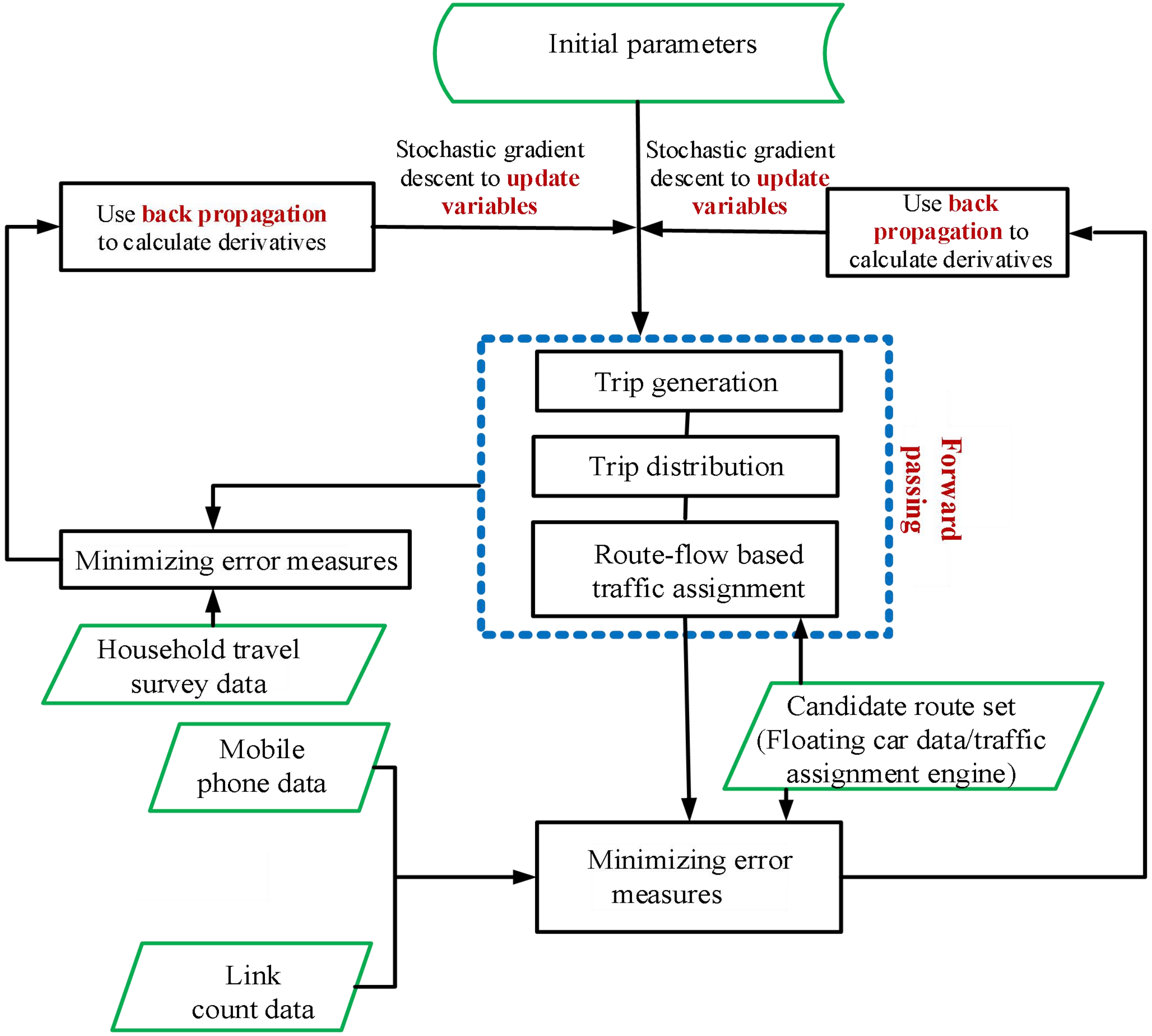
$$(2)\ \frac{\partial F_2(\gamma)}{\partial \gamma}$$

Big data driven transportation computational graph to implement back propagation algorithm

(1) Forward passing
The forward passing process sequentially implements trip generation, trip distribution estimation, and route-based traffic assignment, which is an analogous process of the four-step approach in the field of traffic management.

(2) **Backward propagation**:
The back propagation process inversely implements a feedback control on the forward passing process. Different layers of first-order partial derivatives or "loss errors" are aggregated to calculate marginal gradients.

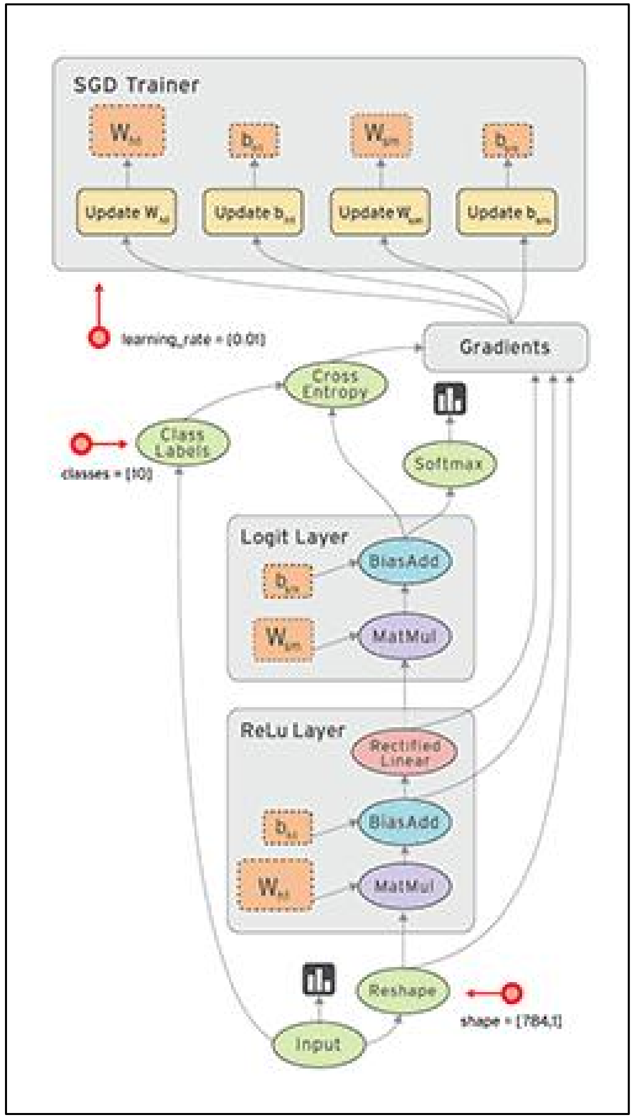
(3) **Variable values update**:
The estimation variables are updated using stochastic gradient descent using the marginal gradients



Arizona State University

Big data driven transportation computational graph (BTCG) to implement back propagation algorithm

- **Automatic differentiation (AD)** is a method to calculate the derivatives of the loss function with respect to the demand variables in a BTCG .
- Derivatives were derived in a **computational graph, using principles of dynamic programming (DP)** (like calculating the shortest path using label correcting algorithm).
- Gradient descent with **Back Propagation (BP) algorithm** guaranteed to find the global minimum of the error function. However, multi-sample-based **stochastic gradient descent** (SGD) can be used to overcome the limit to some extent.
- “AD+DP+BP” can be achieved easily using existing popular data programming tools such as **TensorFlow, Theano** etc.



An illustrative computational graph from:

https://www.tensorflow.org/guide/graphs?hl=zh-cn

Arizona State University

Carefully determine the step sizes in Stochastic Gradient Descent

(1) Vanishing gradient problem in BTCG

$$\frac{\partial F_3(\mathbf{v})}{\partial \boldsymbol{\alpha}} = \frac{\partial F_3(\mathbf{v})}{\partial \mathbf{v}} \times \frac{\partial \mathbf{v}}{\partial \mathbf{f}} \times \frac{\partial \mathbf{f}}{\partial \mathbf{q}} \times \frac{\partial \mathbf{q}}{\partial \boldsymbol{\alpha}}$$

Small values within range [0,1]

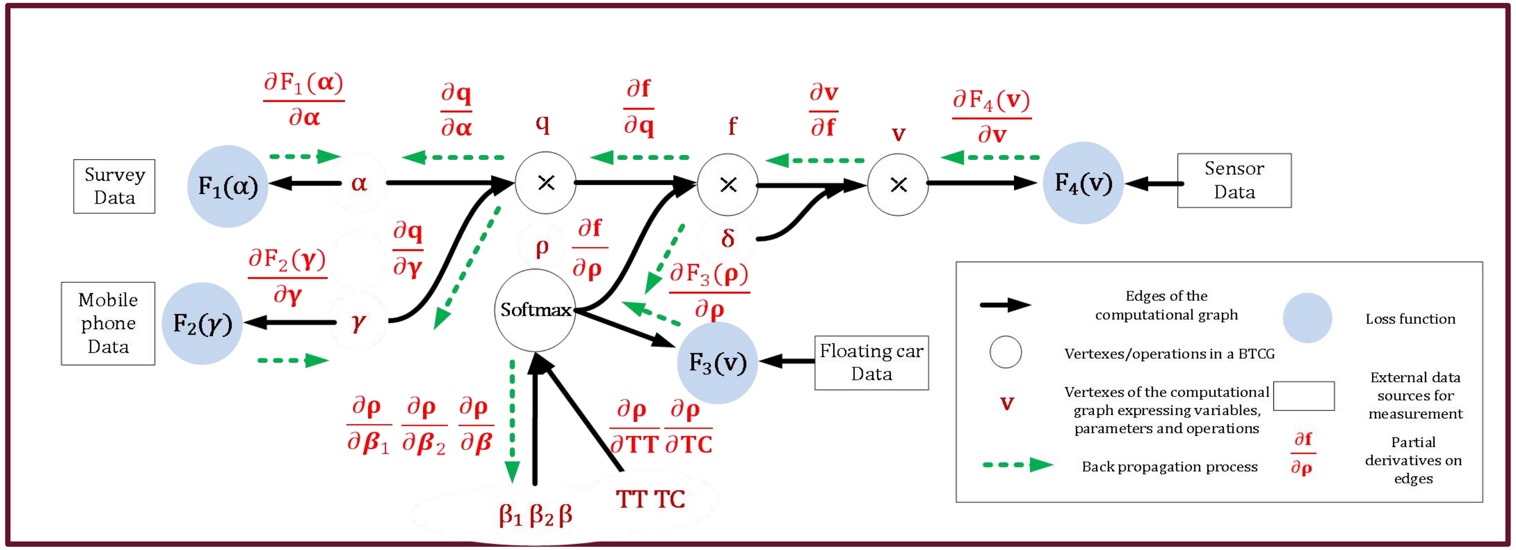
(2) Exploding gradient problem in BTCG

$$\frac{\partial F_4(\mathbf{v})}{\partial \boldsymbol{\gamma}} = \frac{\partial F_3(\mathbf{v})}{\partial \mathbf{v}} \times \frac{\partial \mathbf{v}}{\partial \mathbf{f}} \times \frac{\partial \mathbf{f}}{\partial \mathbf{q}} \times \frac{\partial \mathbf{q}}{\partial \boldsymbol{\gamma}}$$

$$\frac{\partial F_4(\mathbf{v})}{\partial \boldsymbol{\rho}} = \frac{\partial F_3(\mathbf{v})}{\partial \mathbf{v}} \times \frac{\partial \mathbf{v}}{\partial \mathbf{f}} \times \frac{\partial \mathbf{f}}{\partial \boldsymbol{\rho}}$$

Large values equal to OD volume or trip production

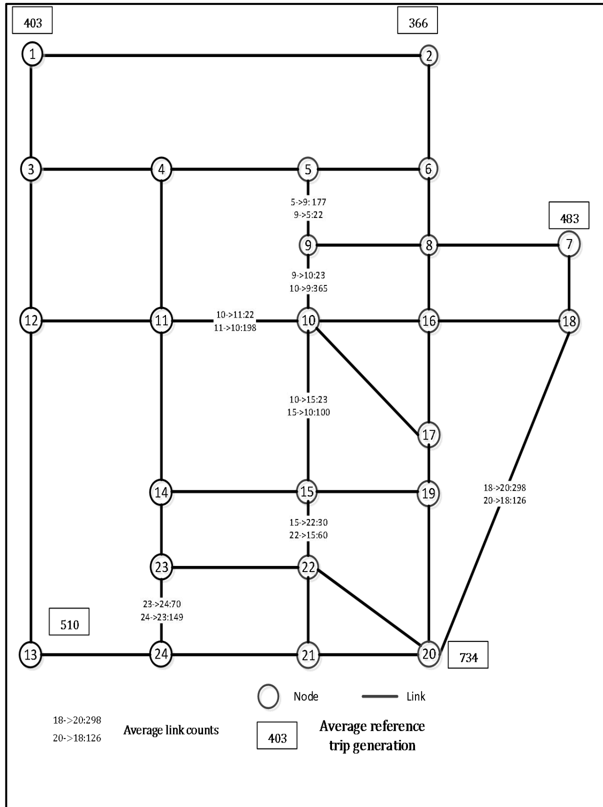
Big data driven Transportation Computational Graph (BTCG)



4. Steps for integrating ML/CG in 4-step model calibration

Numerical Examples

Case study: Sioux Falls network



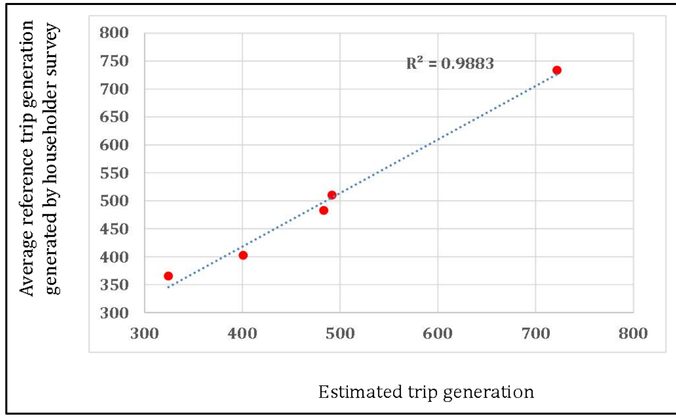
Average reference path proportions

1-> 3-> 4-> 5-> 9:	0.805
1->3->12->11->10->9:	0.195
7->18->20->21->24:	0.4
7->18->20->22->23->25:	0.39
7->8->9->10->11->14->23->24	0.21
20->21->24:	0.51
20->22->23->25:	0.49

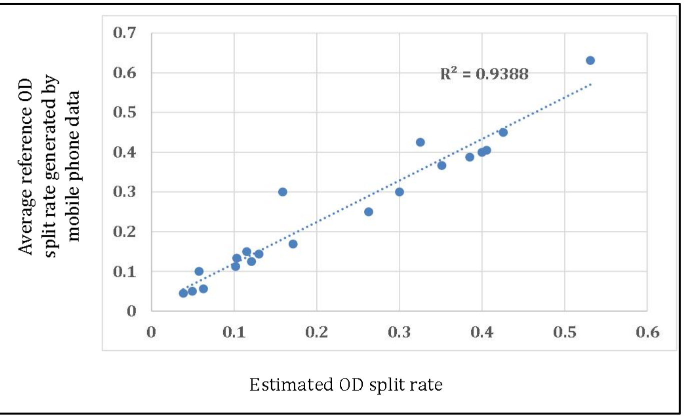
Average reference OD split rate

origin	destination	Average reference OD split rate
1	9	0.14375
1	11	0.05625
1	22	0.63125
1	24	0.16875
2	9	0.1125
2	11	0.3875
2	22	0.133333
2	24	0.366667
7	9	0.45
7	11	0.1
7	22	0.3
7	24	0.15
13	9	0.25
13	11	0.3
13	22	0.05
13	24	0.4
20	9	0.425
20	11	0.045
20	22	0.405
20	24	0.125

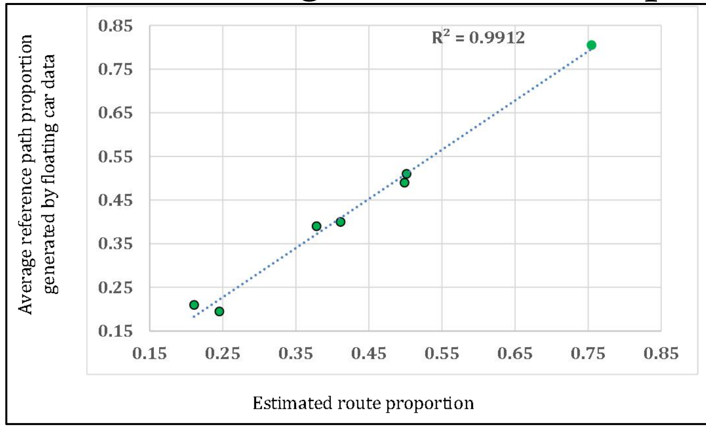
Survey data(5 samples)



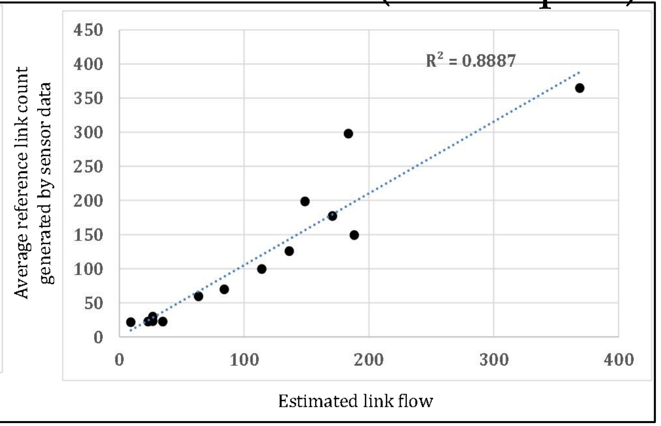
Cell phone data(20 samples)



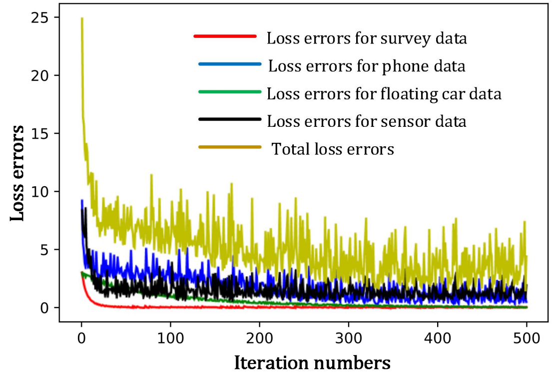
Floating car data(7 samples)



Sensor data(13 samples)



Convergence curves



Convergence curves of the experiments

Arizona State University

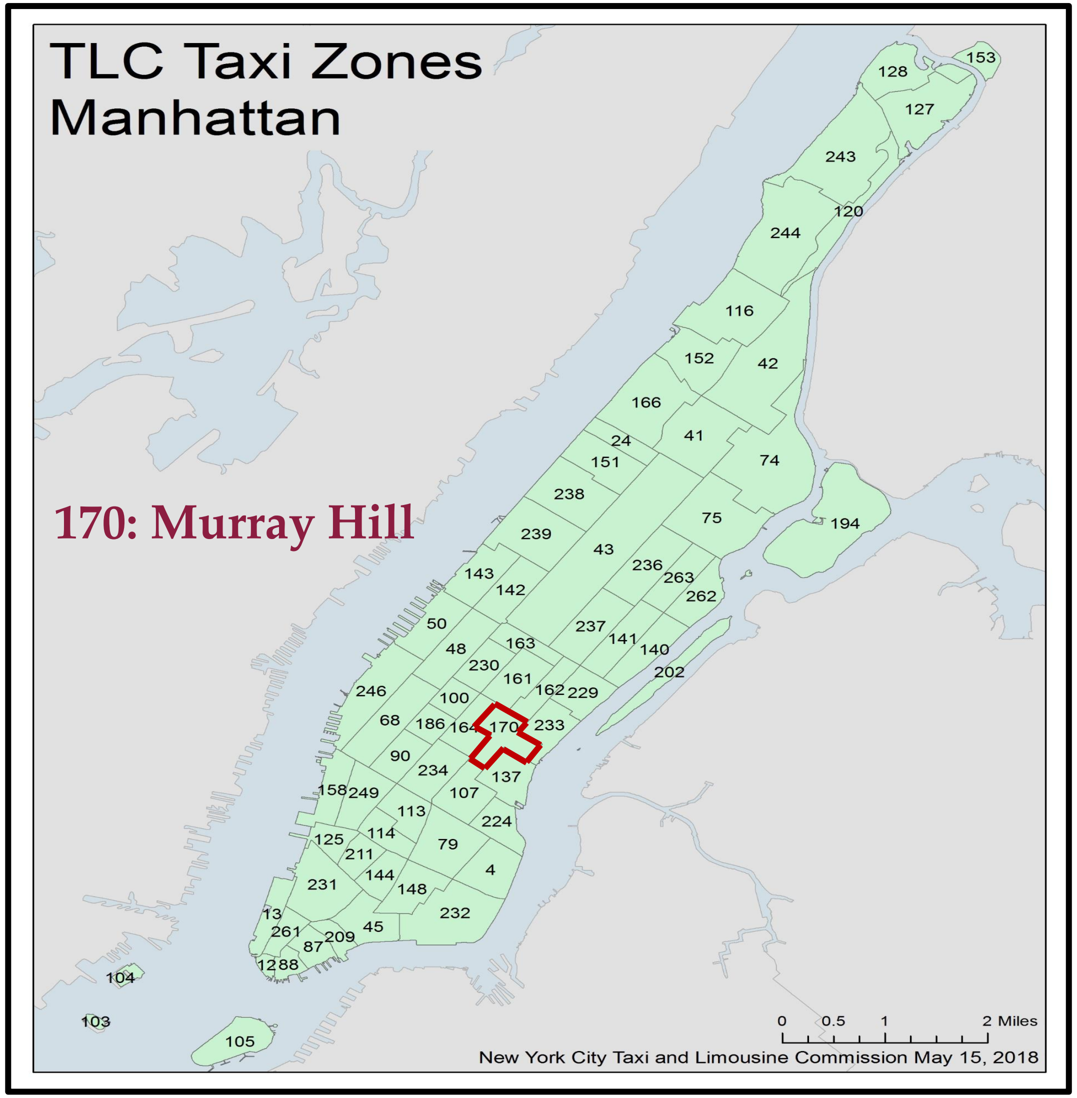
5.Simultaneous Forecasting Using an Econometric Model and Deep Neural Network

Ride-hailing and Existing Mobility Services in New York City

Coauthors: Tae Hooie Kim, Ram Pendyala

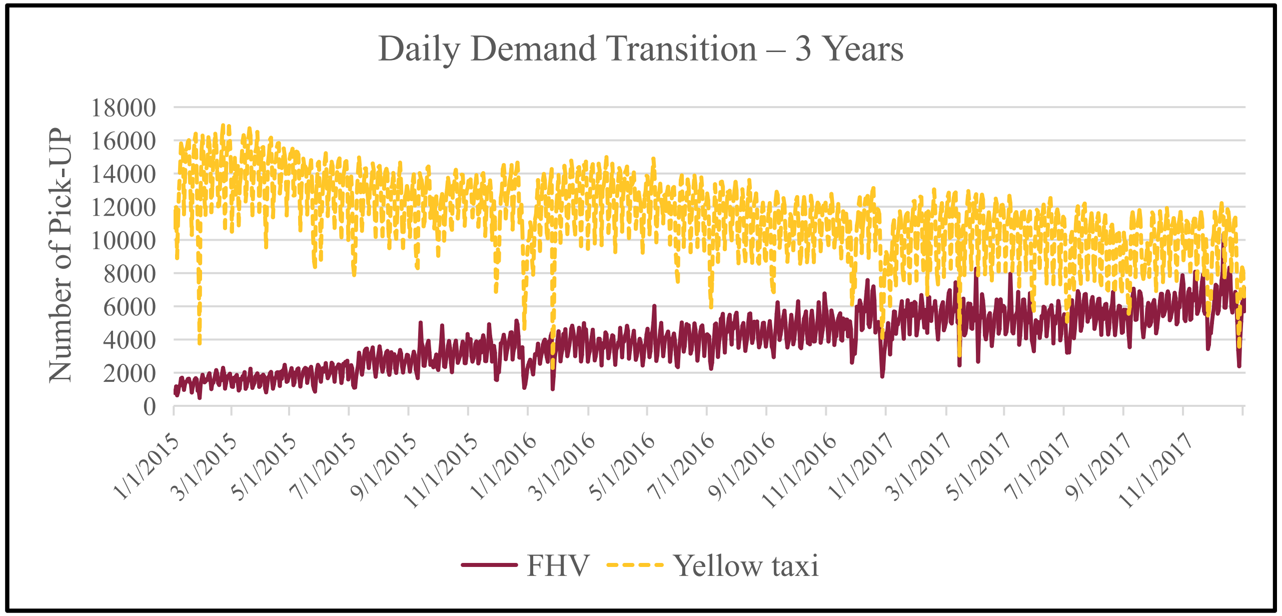
Arizona State University

Motivation: Ride-hailing and Existing Mobility

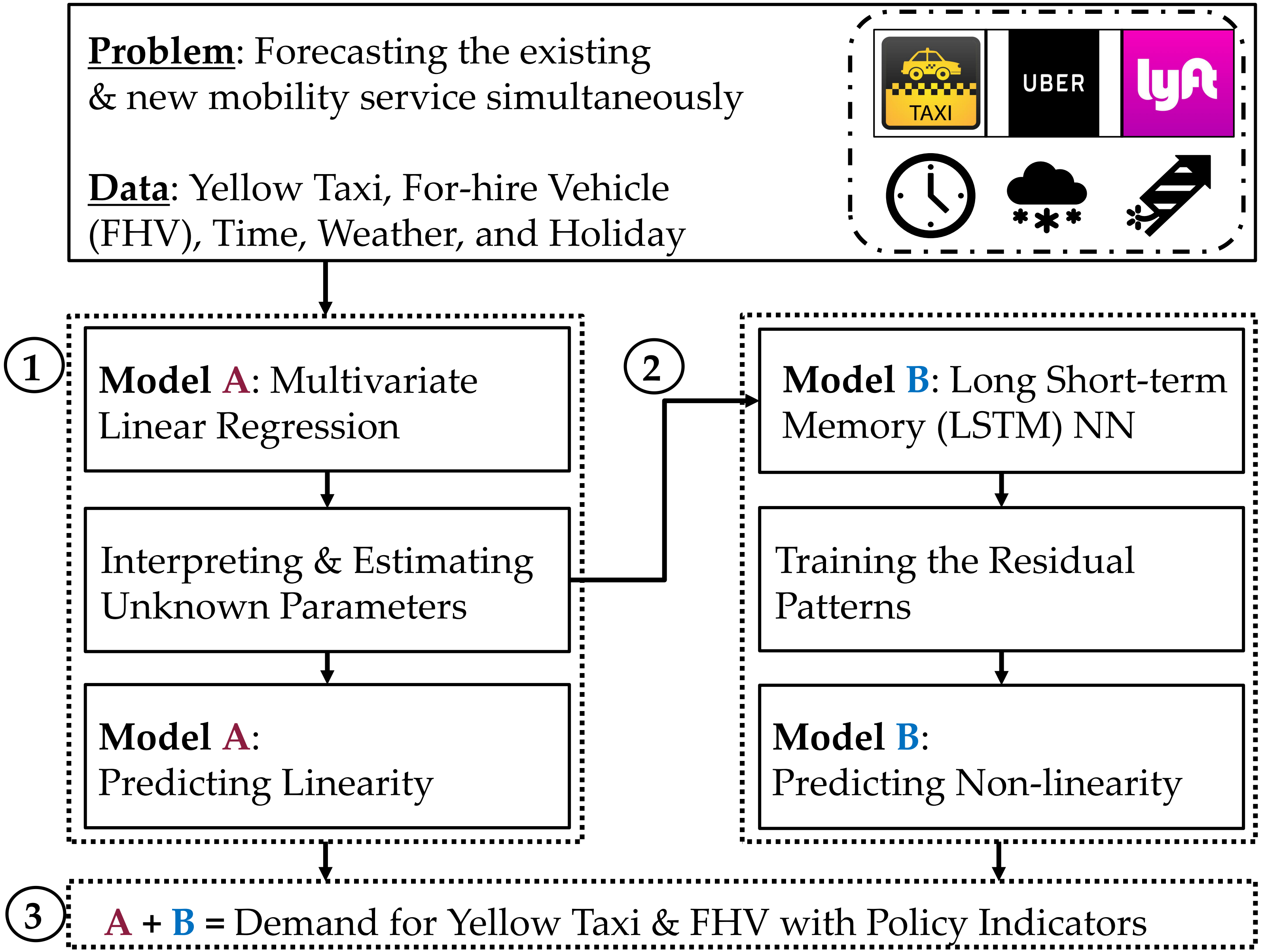


Three Years Total Trips Transition in Murray Hill, Manhattan

Market Segment		
Mobility Service	Yellow Taxi	For-hire Vehicle
2015	84.65 %	15.35 %
2016	73.11 %	26.89 %
2017	**64.19 %**	**35.81 %**



Framework: Simultaneous Forecasting Model



Model A: Multivariate Linear Regression

- The linear regression model can explain the linear relationship, but there is a limitation on capturing non-linear pattern.. (Based on **R-squared,** the model can only cover 37.2% or 57.3%)

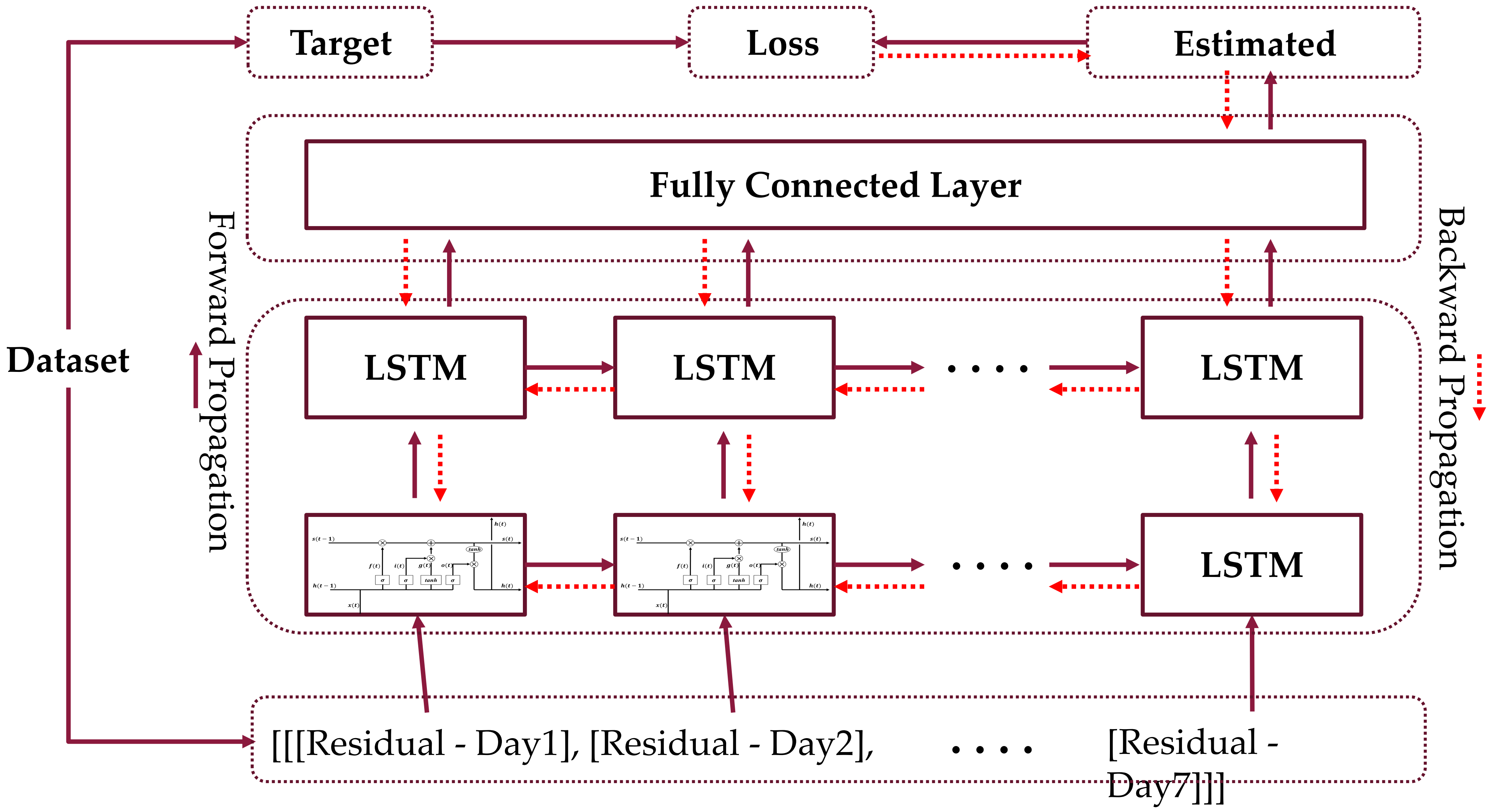
- To cover the residues, we send the data into the Neural Network (**Model B**)

Multivariate Linear Regression (FHV)		
Variables	**Estimate**	**(t-stat)**
Constant	0.6342	(31.418)
Monday	0.0121	(1.212)
Tuesday	0.0943	(9.077)
Wednesday	0.1399	(12.915)
Thursday	0.1995	(17.320)
Friday	0.2089	(18.553)
Saturday	0.0801	(8.111)
Sunday	-0.1005	(-9.859)
Yellow Taxi	-0.6071	(-17.721)
Snow	-0.0349	(-9.062)
Precipitation	0.1833	(4.753)
Holiday	-0.3063	(-10.150)
R-squared	**0.372**	
Observations	931	

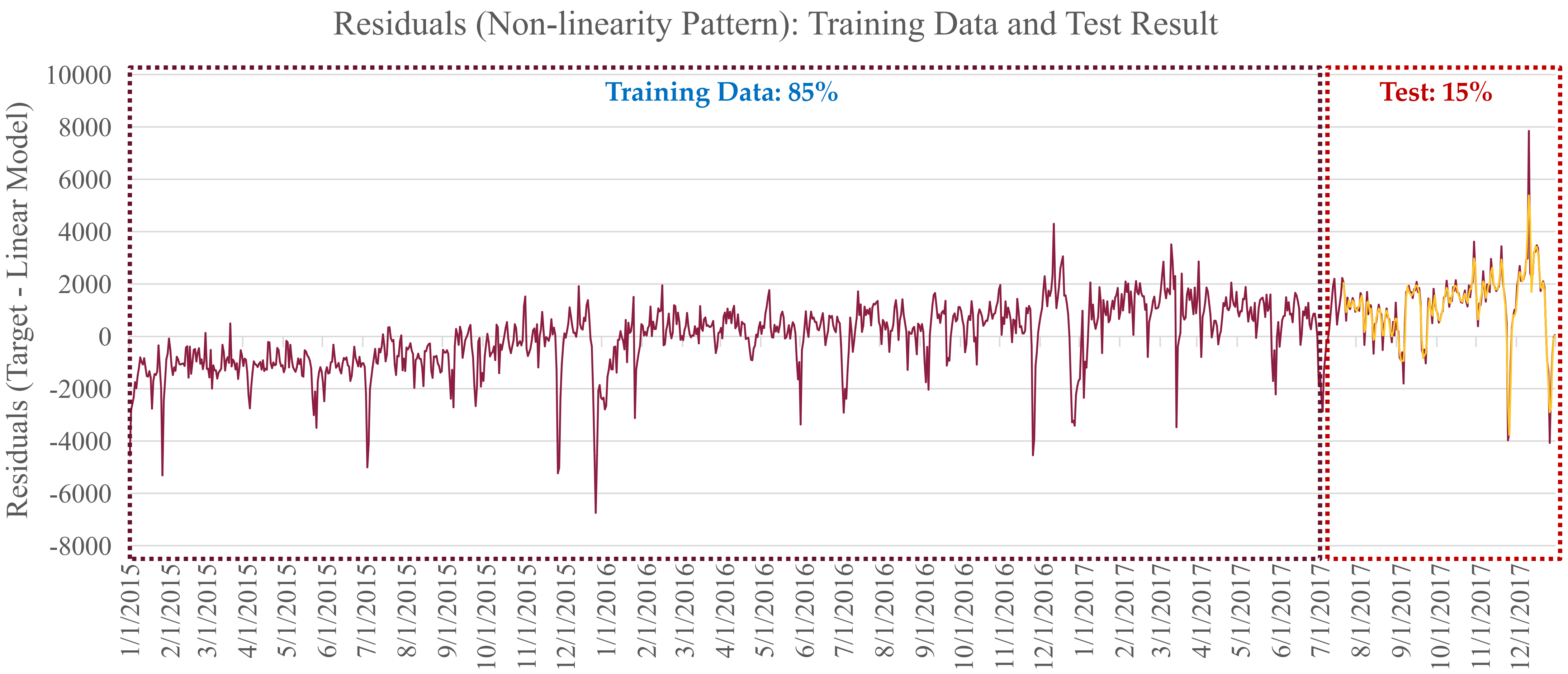
Multivariate Linear Regression (Yellow Taxi)		
Variables	**Estimate**	**(t-stat)**
Constant	0.6975	(94.666)
Monday	0.0302	(3.672)
Tuesday	0.1181	(14.523)
Wednesday	0.1620	(19.768)
Thursday	0.2183	(26.166)
Friday	0.2112	(24.935)
Saturday	0.0666	(8.122)
Sunday	-0.1089	(-13.354)
FHV	-0.4192	(-17.721)
Snow	-0.0343	(-9.06)
Precipitation	-	-
Holiday	-0.3406	(-14.229)
R-squared	**0.573**	
Observations	931	

Unknown Parameters Interpretation			
Explainable Variables		Demand	
		FHV	**Taxi**
Time	Weekday	↑	↑
	Weekend	↓	↓
Weather	Rainy	↑	-
	Snow	↓	↓
Holiday		↓	↓
Relationship		-0.6071 (Yellow)	-0.4192 (FHV)

Model B: Long-short Term Memory Neural Network



Model B: Training Data and Test Data

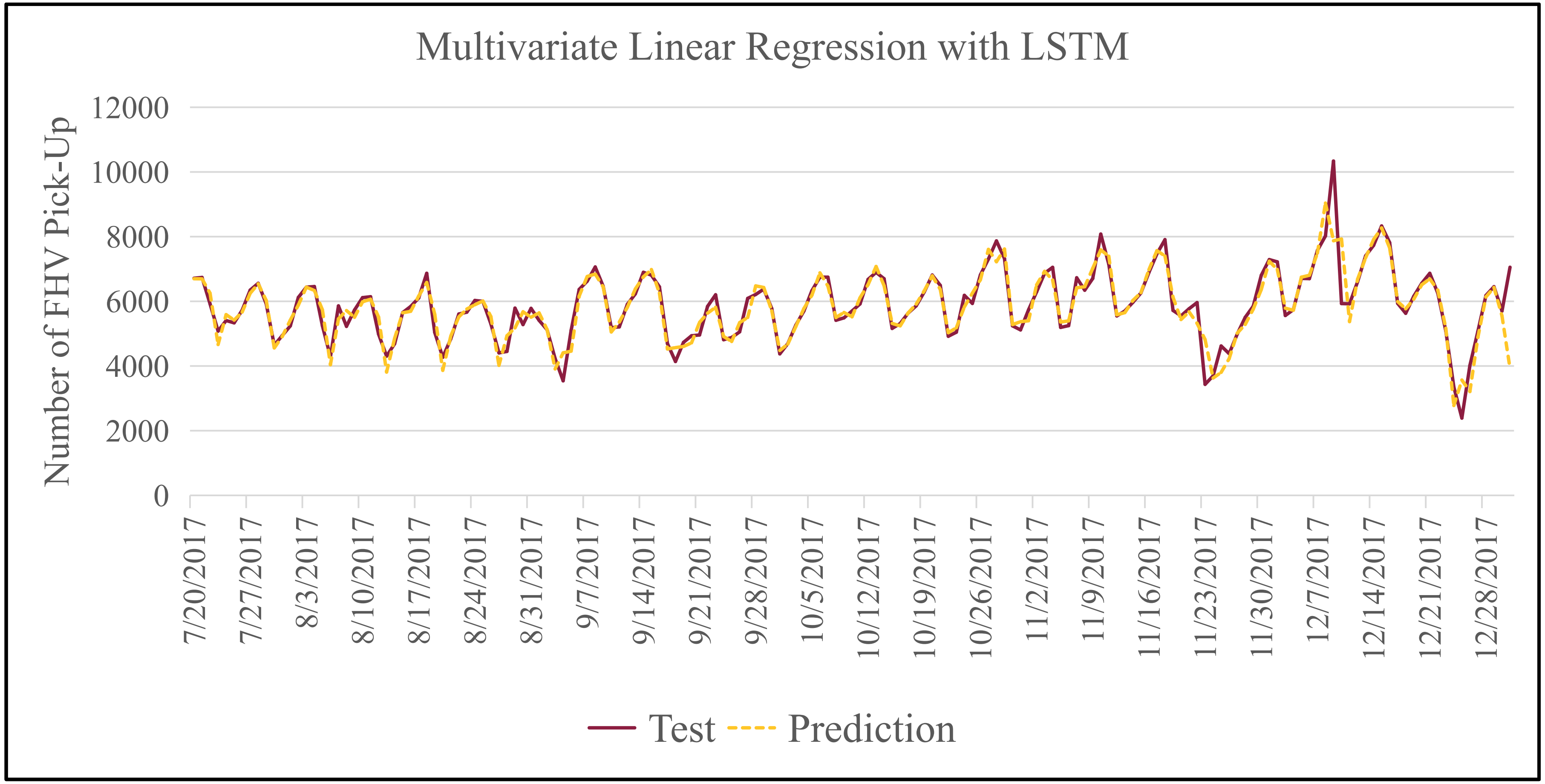


A + B → Forecasting the Demand on FHV

Interpretability

Multivariate Linear Regression (FHV)		
Variables	**Estimate**	**(t-stat)**
Constant	0.6342	(31.418)
Monday	0.0121	(1.212)
Tuesday	0.0943	(9.077)
Wednesday	0.1399	(12.915)
Thursday	0.1995	(17.320)
Friday	0.2089	(18.553)
Saturday	0.0801	(8.111)
Sunday	-0.1005	(-9.859)
Yellow Taxi	-0.6071	(-17.721)
Snow	-0.0349	(-9.062)
Precipitation	0.1833	(4.753)
Holiday	-0.3063	(-10.150)
R-squared	0.372	
Observations	931	

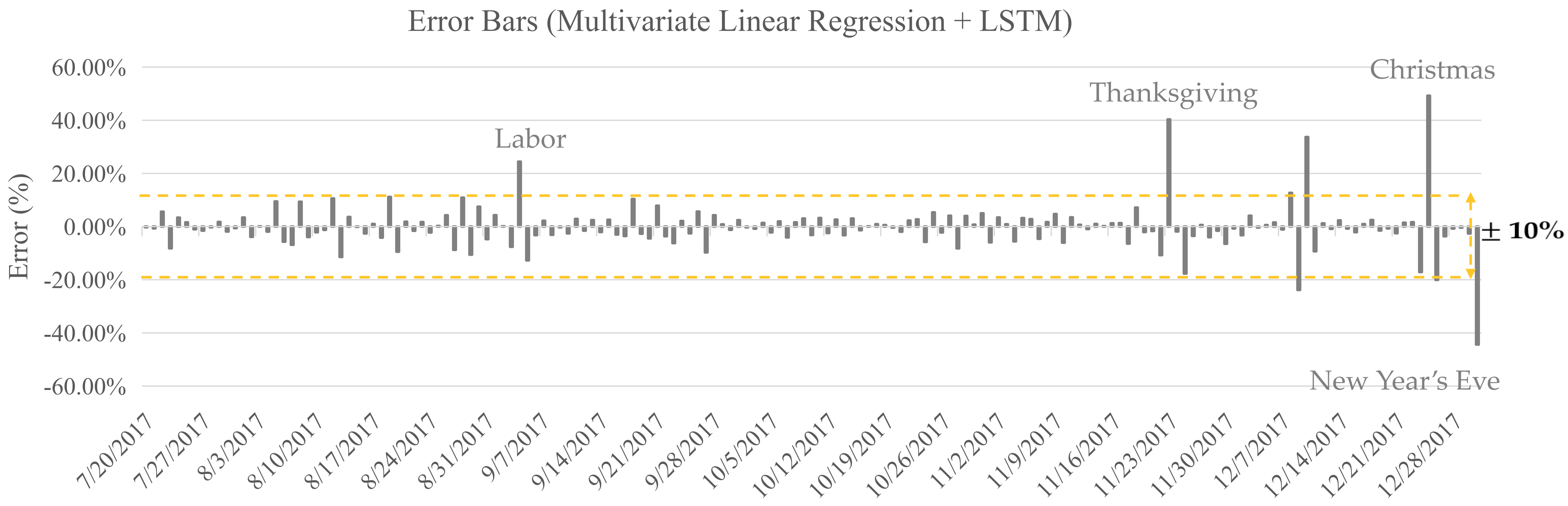
Predictability



Test Data

A+B Model

Model Performance and Comparison



	ARIMA	**LSTM**	**ARIMA+LSTM**	**MLR + LSTM (A+B)**
RMSE	663.1348	460.1567	353.5828	467.3608
MAPE	8.1178	5.8335	4.5204	4.8624
R-squared	0.5915	0.7652	0.8520	0.8060
Min Error(%)	-26.06	-20.25	-20.91	-23.23
Max Error(%)	86.85	63.04	47.58	49.83

6.Further extensions

Further extensions

1. Extend our computational graph framework for **Multi-day learning and dynamic traffic systems.**

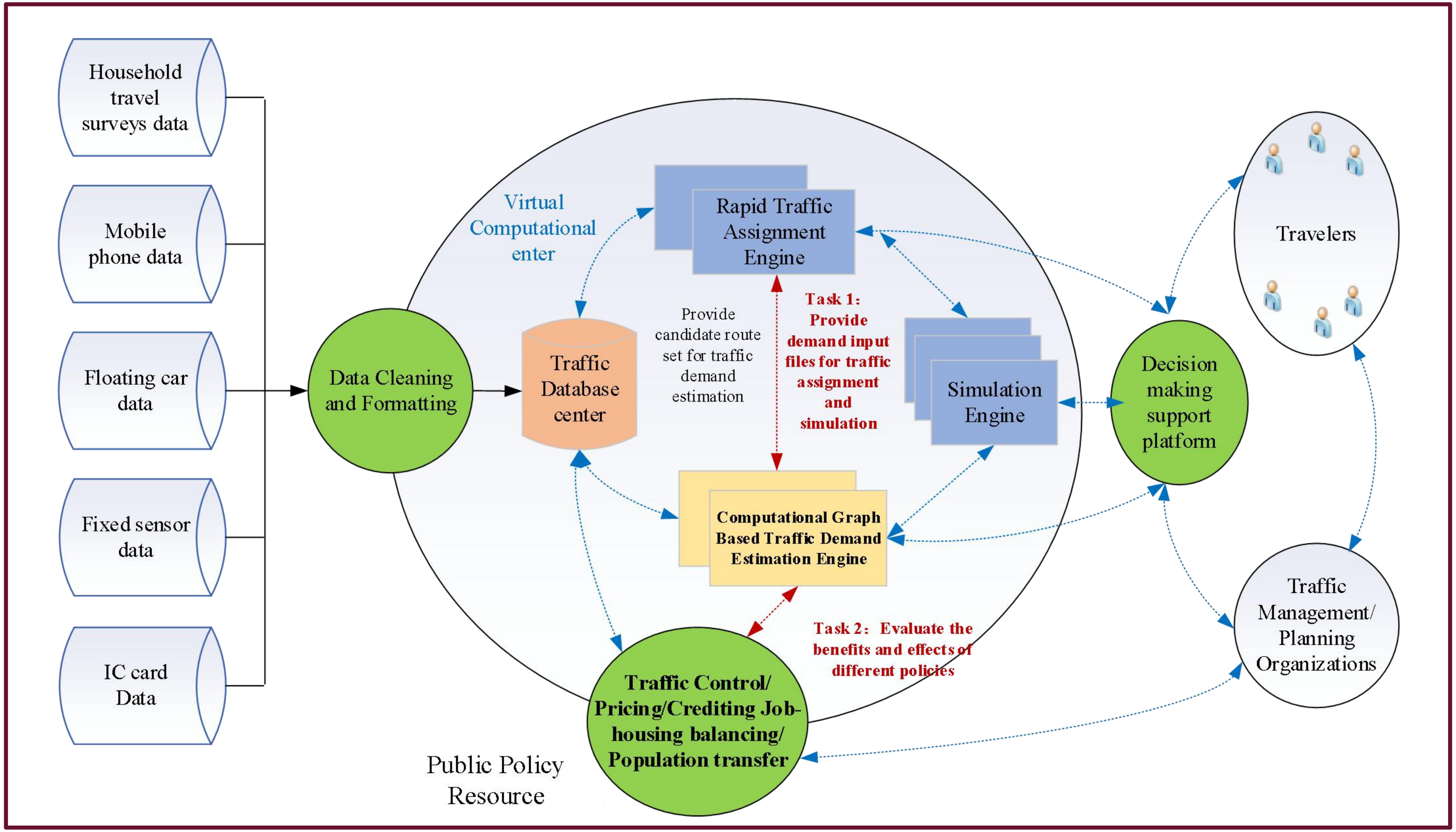
2. Extend our computational graph framework to **activity-based models.**

3. Capture **behavioral** causes by incorporating more complex choice models.

4. Improve computational capability using distributed computational graph to speed up the algorithm.

Further extensions

Data-driven traffic decision-making and management system for departments of transportation (DOT) and Metropolitan Planning Organizations (MPO) in future



Our paper:

Wu, X., Guo, J., Xian, K., & Zhou, X. (2018). Hierarchical travel demand estimation using multiple data sources: A forward and backward propagation algorithmic framework on a layered computational graph. *Transportation Research Part C: Emerging Technologies*, *96*, 321-346.

https://www.researchgate.net/publication/325131295_Hierarchical_travel_demand_estimation_using_multiple_data_sources_A_forward_and_backward_propagation_algorithmic_framework_on_a_layered_computational_graph

Our codes:

For educational and research purposes, one can find the Matlab and Python source code for small networks at https://github.com/xzhou99/BTCG.

Supported from NSF CMMI#**1538569, Improving Spatial Observability of Dynamic Traffic Systems**

Thanks

Questions?