Lecture 6: Electronic Health Records (Part 2)

Announcements

Upcoming deadlines:

- A1 due Tue 10/18
- Project proposal due Fri 10/21
  - Remember that you must **train** a deep learning model somewhere in your project!
- Project partner finding session during review section this Friday, 1:30pm, Alway M106

Agenda for today

- Finishing up from last time: RNN (LSTM) models for EHR prediction tasks
- More on EHR data
- More on feature representations
- A first look at model interpretability: soft attention

Last time: overview of electronic health records

Patient chart in digital form, containing medical and treatment history

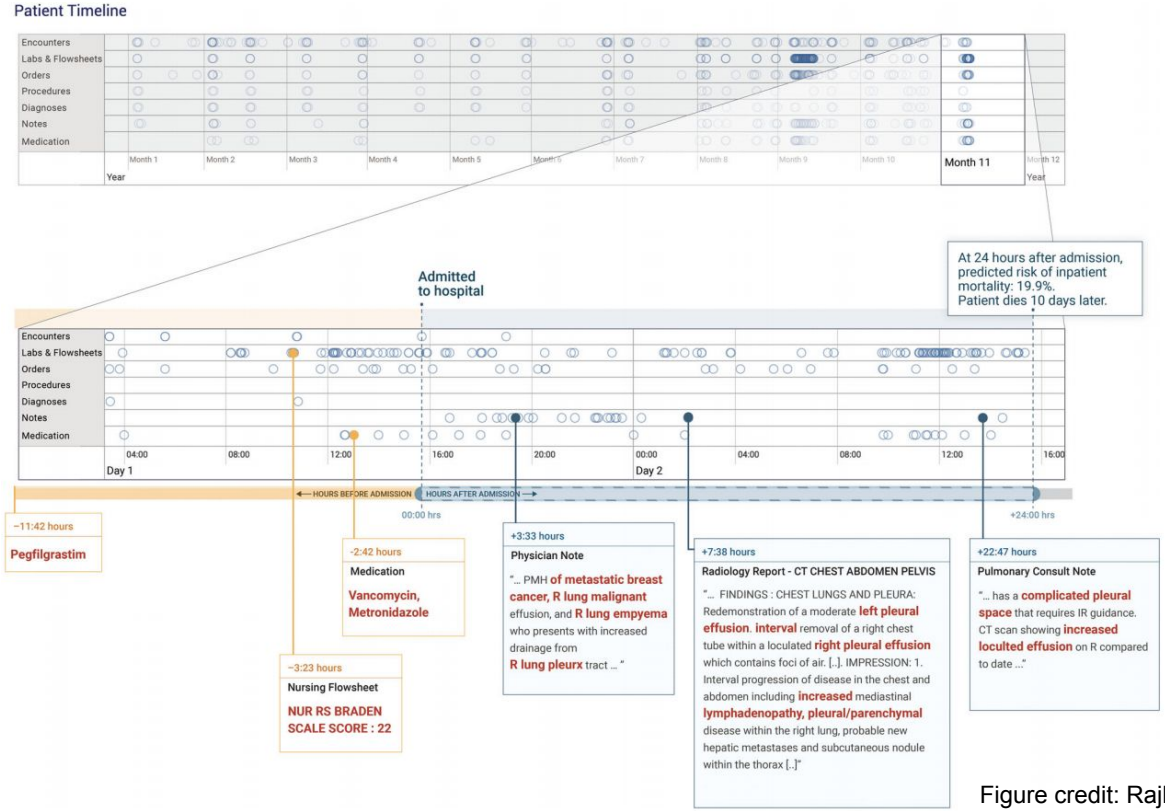
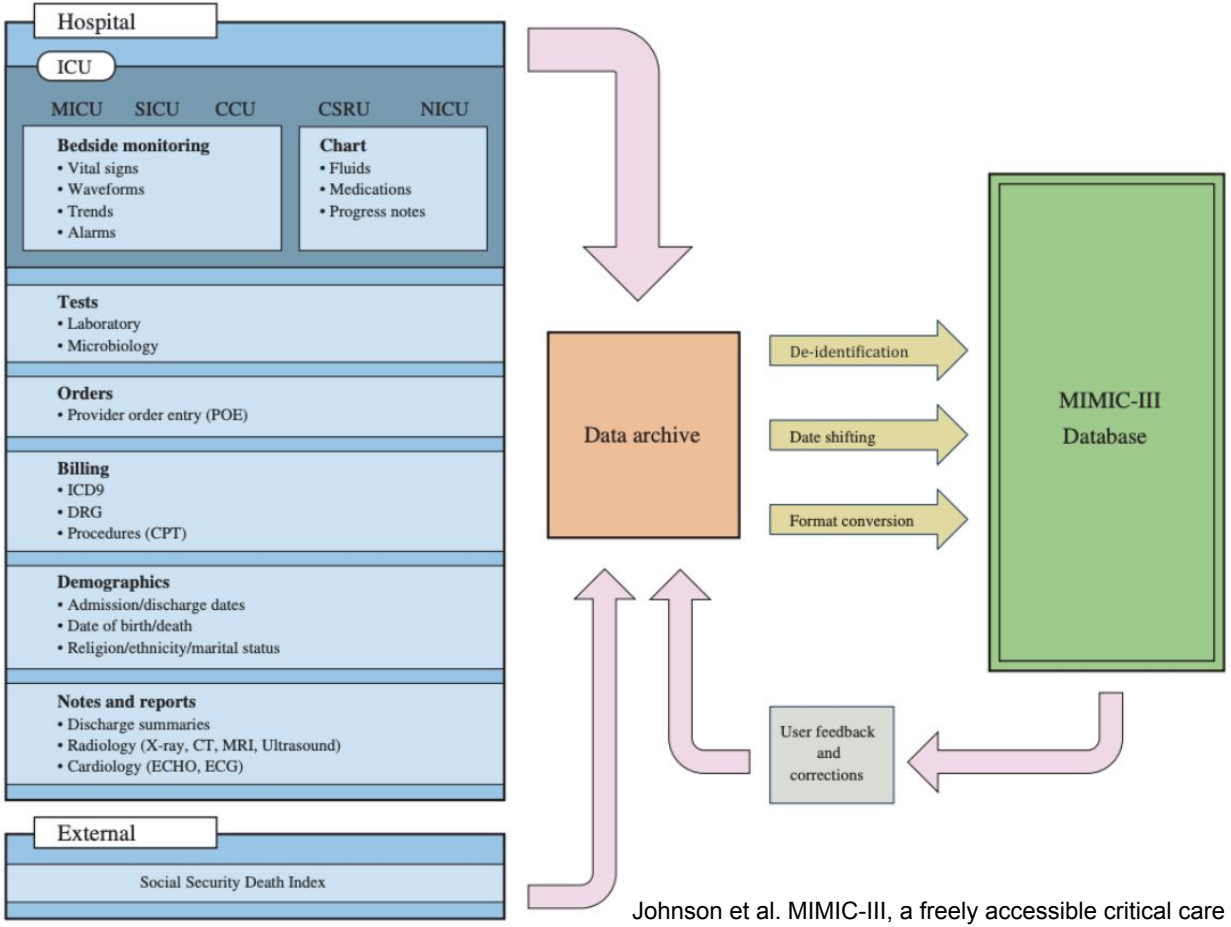


Figure credit: Rajkomar et al. 2018

A real example of EHR data: MIMIC-III dataset

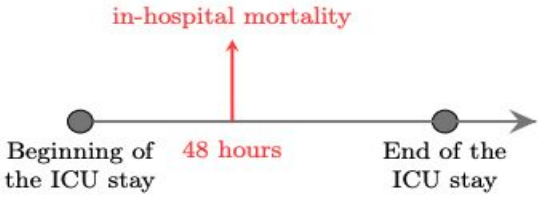


Johnson et al. MIMIC-III, a freely accessible critical care database. 2016.

Examples of prediction tasks

In-hospital mortality

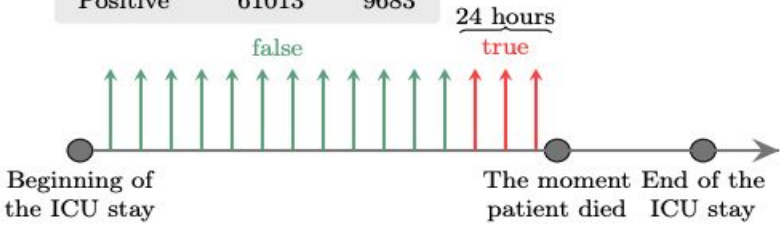
	Train	Test
Negative	15480	2862
Positive	2423	374



(a)

Decompensation

	Train	Test
Negative	2847401	513525
Positive	61013	9683



(b)

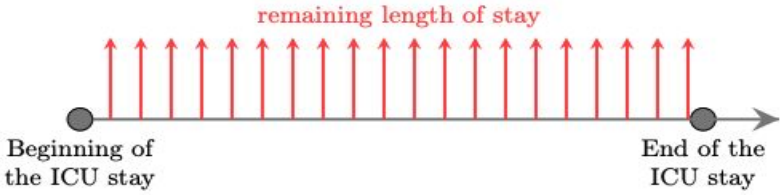
Phenotypes

Train	Test
35621	6281

(c)

Length-of-stay

Train	Test
2925434	525912



(d)

Harutyunyan et al. 2019

Remember: “vanilla” neural networks for predictions from clinical variables

Let us consider the task of **regression**: predicting a single real-valued output from input data

Model input: data vector $x = [x_1, x_2, ..., x_N]$

Model output: prediction (single number) $\hat{y}$

Example: predicting hospital length-of-stay from clinical variables in the electronic health record

$x =$ [age, weight, …, temperature, oxygen saturation]

$\hat{y} =$ length-of-stay (days)

Recurrent Neural Network

We can process a sequence of vectors **x** by applying a **recurrence formula** at every time step:

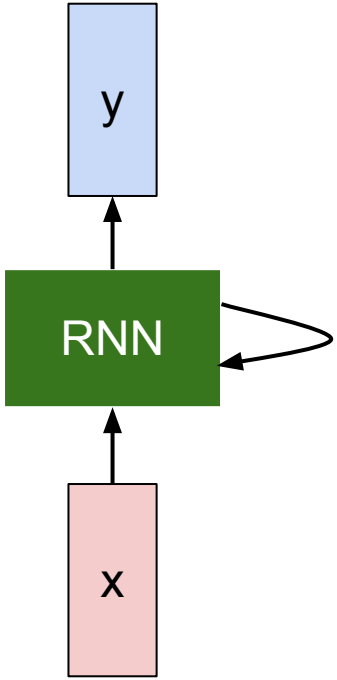
$$\boxed{h_t} = \boxed{f_W}(\boxed{h_{t-1}}, \boxed{x_t})$$

new state

old state

input vector at some time step

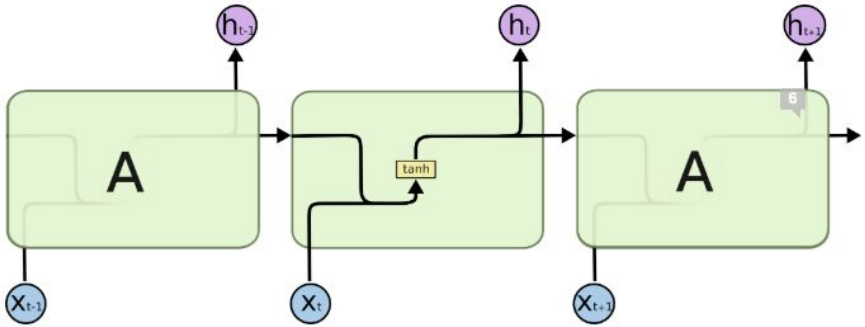
some function with parameters W



Slide credit: CS231n

Long Short Term Memory (LSTM) Recurrent Networks

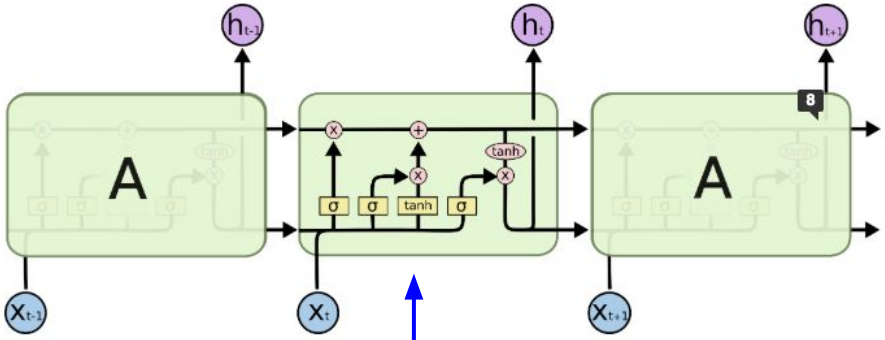
Unrolled Vanilla RNN



$$h_t = \tanh(W_{hh} h_{t-1} + W_{xh} x_t)$$

$$y_t = W_{hy} h_t$$

Unrolled LSTM



Different computation to obtain h_t

Figure credit: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Harutyunyan et al.

- Benchmarked LSTMs vs logistic regression on common prediction tasks using MIMIC-III data
- In-hospital mortality, decompensation, length-of-stay, phenotype classification
- Used a subset of 17 clinical variables from MIMIC-III

Variable	MIMIC-III table	Impute value	Modeled as
Capillary refill rate	chartevents	0.0	categorical
Diastolic blood pressure	chartevents	59.0	continuous
Fraction inspired oxygen	chartevents	0.21	continuous
Glascow coma scale eye opening	chartevents	4 spontaneously	categorical
Glascow coma scale motor response	chartevents	6 obeys commands	categorical
Glascow coma scale total	chartevents	15	categorical
Glascow coma scale verbal response	chartevents	5 oriented	categorical
Glucose	chartevents, labevents	128.0	continuous
Heart Rate	chartevents	86	continuous
Height	chartevents	170.0	continuous
Mean blood pressure	chartevents	77.0	continuous
Oxygen saturation	chartevents, labevents	98.0	continuous
Respiratory rate	chartevents	19	continuous
Systolic blood pressure	chartevents	118.0	continuous
Temperature	chartevents	36.6	continuous
Weight	chartevents	81.0	continuous
pH	chartevents, labevents	7.4	continuous

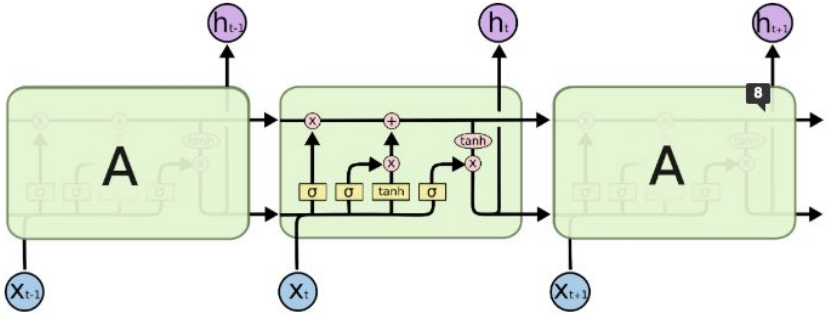
Harutyunyan et al. Multitask learning and benchmarking with clinical time series data. 2019.

Harutyunyan et al.

- **Logistic regression models**
  - Use hand-engineered feature vector to represent a time-series: min, max, mean, std dev, etc. of each feature in several subsequences (full series, first 10% of series, first 50%, last 10%, etc.)
  - If feature does not occur in subsequence (missing data), impute with mean value from training set
  - Categorical variables had meaningful numeric values -> no change
  - Zero-mean unit-variance standardization of all features

Harutyunyan et al. Multitask learning and benchmarking with clinical time series data. 2019.

Harutyunyan et al.



- **LSTM models**
  - Bucket time series into regularly spaced intervals, take the value (or last value, if multiple) of each variable in the interval to create observation x_t
  - Encode categorical variables using a one-hot vector (vector of 0s with a 1 in the observed position).
  - If variable is missing in a time bucket, impute using most recent observed measurement if it exists, and mean value from training set otherwise
  - Concat the values of each clinical variable with a binary mask indicating presence or not (i.e., missing and needed to impute) to form full observation feature vector x_t

Harutyunyan et al. Multitask learning and benchmarking with clinical time series data. 2019.

Harutyunyan et al.: logistic regression vs LSTMs

Found better performance overall for LSTMs (S) vs logistic regression (LR). Also introduced more sophisticated variants and multi-task training (joint training of all tasks together).

In-hospital Mortality

Model	AUC-ROC
SAPS	0.720 (0.720, 0.720)
APS-III	0.750 (0.750, 0.750)
OASIS	0.760 (0.760, 0.761)
SAPS-II	0.777 (0.776, 0.777)
LR	0.848 (0.828, 0.868)
S	0.855 (0.835, 0.873)
S + DS	0.856 (0.836, 0.875)
C	0.862 (0.844, 0.881)
C + DS	0.854 (0.834, 0.873)
MS	0.861 (0.842, 0.878)
MC	0.870 (0.852, 0.887)

Phenotyping

Model	Macro AUC-ROC
LR	0.739 (0.734, 0.743)
S	0.770 (0.766, 0.775)
S + DS	0.774 (0.769, 0.778)
C	0.776 (0.772, 0.781)
C + DS	0.773 (0.769, 0.777)
MS	0.768 (0.763, 0.772)
MC	0.774 (0.770, 0.778)

LR – logistic regression
S – standard LSTM
C – channel-wise LSTM
DS – deep supervision
MS – multitask standard LSTM
MC – multitask channel-wise LSTM

Figure credit: Harutyunyan et al. Multitask learning and benchmarking with clinical time series data. 2019.

Harutyunyan et al.: logistic regression vs LSTMs

Found better performance overall for LSTMs (S) vs logistic regression (LR). Also introduced more sophisticated variants and multi-task training (joint training of all tasks together).

In-hospital Mortality

Model	AUC-ROC
SAPS	0.720 (0.720, 0.720)
APS-III	0.750 (0.750, 0.750)
OASIS	0.760 (0.760, 0.761)
SAPS-II	0.777 (0.776, 0.777)
LR	0.848 (0.828, 0.868)
S	0.855 (0.835, 0.873)
S + DS	0.856 (0.836, 0.875)
C	0.862 (0.844, 0.881)
C + DS	0.854 (0.834, 0.873)
MS	0.861 (0.842, 0.878)
MC	0.870 (0.852, 0.887)

Phenotyping

Model	Macro AUC-ROC
LR	0.739 (0.734, 0.743)
S	0.770 (0.766, 0.775)
S + DS	0.774 (0.769, 0.778)
C	0.776 (0.772, 0.781)
C + DS	0.773 (0.769, 0.777)
MS	0.768 (0.763, 0.772)
MC	0.774 (0.770, 0.778)

LR – logistic regression
S – standard LSTM
C – channel-wise LSTM
DS – deep supervision
MS – multitask standard LSTM
MC – multitask channel-wise LSTM

Found better performance for phenotyping acute vs chronic conditions -- makes sense!

Figure credit: Harutyunyan et al. Multitask learning and benchmarking with clinical time series data. 2019.

Recall: Harutyunyan et al. imputed missing data

- **Logistic regression models**
  - Use hand-engineered feature vector to represent a time-series: min, max, mean, std dev, etc. of each feature in several subsequences (full series, first 10% of series, first 50%, last 10%, etc.)
  - If feature does not occur in subsequence (missing data), impute with mean value from training set
  - Categorical variables had meaningful numeric values -> no change
  - Zero-mean unit-variance standardization of all features

Harutyunyan et al. Multitask learning and benchmarking with clinical time series data. 2019.

More on missing data

A common problem with clinical variable data

- Missing completely at random (MCAR)
  - Missingness does not depend on the missing variable or on other variables
  - Ex: A portion of patient pain surveys (producing variable of patient pain level) are randomly lost or unreadable
- Missing at random (MAR)
  - Missingness does not depend on the missing variable but may depend on other variables
  - Ex: Male patients are less likely to complete patient pain surveys
- Missing not at random (MNAR)
  - Missingness can depend on the missing variable itself
  - Ex: Patients with higher pain levels are less likely to complete patient pain surveys

More on missing data

A common problem with clinical variable data

- Missing completely at random (MCAR)
  - Missingness does not depend on the missing variable or on other variables
  - Ex: A portion of patient pain surveys (producing variable of patient pain level) are randomly lost or unreadable
- Missing at random (MAR)
  - Missingness does not depend on the missing variable but may depend on other variables
  - Ex: Male patients are less likely to complete patient pain surveys
- Missing not at random (MNAR)
  - Missingness can depend on the missing variable itself
  - Ex: Patients with higher pain levels are less likely to complete patient pain surveys

MNAR highest degree of bias / most challenging to accurately impute. Analysis of how well imputation methods work for MCAR / MAR / MNAR cases beyond the scope of this course -> just know that these are missingness characteristics that can make accurate imputation more or less challenging.

Strategies to impute data

- Simplest approaches:
  - Delete records with missing data
  - Fixed imputation of missing values with mean, median, previous value, interpolation, etc.

Strategies to impute data

- Simplest approaches:
  - Delete records with missing data
  - Fixed imputation of missing values with mean, median, previous value, interpolation, etc.
- More sophisticated approaches:
  - K-nearest neighbors (impute based on feature value of k closest neighbors determined through non-missing values)
  - Predicting missing values (single imputation): Train regression or classification models to predict missing values based on other variables

Strategies to impute data

- Simplest approaches:
  - Delete records with missing data
  - Fixed imputation of missing values with mean, median, previous value, interpolation, etc.
- More sophisticated approaches:
  - K-nearest neighbors (impute based on feature value of k closest neighbors determined through non-missing values)
  - Predicting missing values (single imputation): Train regression or classification models to predict missing values based on other variables
- Even more sophisticated approaches:
  - Predicting missing values (multiple imputation): Perform single imputation multiple times based on different random initializations, then aggregate for final imputation + uncertainty measurement

Strategies to impute data

- Simplest approaches:
  - Delete records with missing data
  - Fixed imputation of missing values with mean, median, previous value, interpolation, etc.
- More sophisticated approaches:
  - K-nearest neighbors (impute based on feature value of k closest neighbors determined through non-missing values)
  - Predicting missing values (single imputation): Train regression or classification models to predict missing values based on other variables
- Even more sophisticated approaches:
  - Predicting missing values (multiple imputation): Perform single imputation multiple times based on different random initializations, then aggregate for final imputation + uncertainty measurement
- An ongoing active area of research:
  - Methods incorporating deep learning generative models, etc.

Example of imputation through prediction in the widely used MICE Algorithm[1]

A	B	C
0.93	1.40	1.53
0.24	0.46	0.76
	0.80	
0.95	1.24	1.46
0.23	0.57	
0.90		1.28
0.15	0.42	
0.47	0.54	0.63
	1.14	
0.89	1.23	1.45

Red = missing values across features A,B,C

[1]van Burren, 2011. Figure credit: https://cran.r-project.org/web/packages/miceRanger/vignettes/miceAlgorithm.html

Example of imputation through prediction in the widely used MICE Algorithm[1]

A	B	C
0.93	1.40	1.53
0.24	0.46	0.76
	0.80	
0.95	1.24	1.46
0.23	0.57	
0.90		1.28
0.15	0.42	
0.47	0.54	0.63
	1.14	
0.89	1.23	1.45

A	B	C
0.93	1.40	1.53
0.24	0.46	0.76
0.90	0.80	1.53
0.95	1.24	1.46
0.23	0.57	1.28
0.90	0.46	1.28
0.15	0.42	1.53
0.47	0.54	0.63
0.47	1.14	1.28
0.89	1.23	1.45

Fill in missing entries with initial values (random, means, randomly drawn from distribution, etc.)

[1]van Burren, 2011. Figure credit: https://cran.r-project.org/web/packages/miceRanger/vignettes/miceAlgorithm.html

Example of imputation through prediction in the widely used MICE Algorithm[1]

A	B	C
0.93	1.40	1.53
0.24	0.46	0.76
	0.80	
0.95	1.24	1.46
0.23	0.57	
0.90		1.28
0.15	0.42	
0.47	0.54	0.63
	1.14	
0.89	1.23	1.45

A	B	C
0.93	1.40	1.53
0.24	0.46	0.76
0.90	0.80	1.53
0.95	1.24	1.46
0.23	0.57	1.28
0.90	0.46	1.28
0.15	0.42	1.53
0.47	0.54	0.63
0.47	1.14	1.28
0.89	1.23	1.45

A	B	C
0.93	1.40	1.53
0.24	0.46	0.76
0.24	0.80	1.53
0.95	1.24	1.46
0.23	0.57	1.28
0.90	0.46	1.28
0.15	0.42	1.53
0.47	0.54	0.63
0.89	1.14	1.28
0.89	1.23	1.45

Update missing values for feature A using regression model trained on all values (including red) of other features

[1]van Burren, 2011. Figure credit: https://cran.r-project.org/web/packages/miceRanger/vignettes/miceAlgorithm.html

Example of imputation through prediction in the widely used MICE Algorithm[1]

A	B	C
0.93	1.40	1.53
0.24	0.46	0.76
	0.80	
0.95	1.24	1.46
0.23	0.57	
0.90		1.28
0.15	0.42	
0.47	0.54	0.63
	1.14	
0.89	1.23	1.45

A	B	C
0.93	1.40	1.53
0.24	0.46	0.76
0.90	0.80	1.53
0.95	1.24	1.46
0.23	0.57	1.28
0.90	0.46	1.28
0.15	0.42	1.53
0.47	0.54	0.63
0.47	1.14	1.28
0.89	1.23	1.45

A	B	C
0.93	1.40	1.53
0.24	0.46	0.76
0.24	0.80	1.53
0.95	1.24	1.46
0.23	0.57	1.28
0.90	0.46	1.28
0.15	0.42	1.53
0.47	0.54	0.63
0.89	1.14	1.28
0.89	1.23	1.45

A	B	C
0.93	1.40	1.53
0.24	0.46	0.76
0.24	0.80	1.53
0.95	1.24	1.46
0.23	0.57	1.28
0.90	1.24	1.28
0.15	0.42	1.53
0.47	0.54	0.63
0.89	1.14	1.28
0.89	1.23	1.45

Update missing values for feature B using regression model trained on all values (including red and updated/yellow) of other features

[1]van Burren, 2011. Figure credit: https://cran.r-project.org/web/packages/miceRanger/vignettes/miceAlgorithm.html

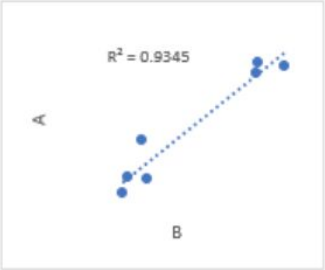
Example of imputation through prediction in the widely used MICE Algorithm[1]

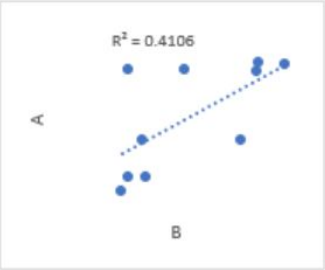
A	B	C
0.93	1.40	1.53
0.24	0.46	0.76
	0.80	
0.95	1.24	1.46
0.23	0.57	
0.90		1.28
0.15	0.42	
0.47	0.54	0.63
	1.14	
0.89	1.23	1.45

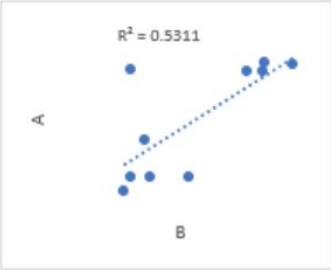
A	B	C
0.93	1.40	1.53
0.24	0.46	0.76
0.90	0.80	1.53
0.95	1.24	1.46
0.23	0.57	1.28
0.90	0.46	1.28
0.15	0.42	1.53
0.47	0.54	0.63
0.47	1.14	1.28
0.89	1.23	1.45

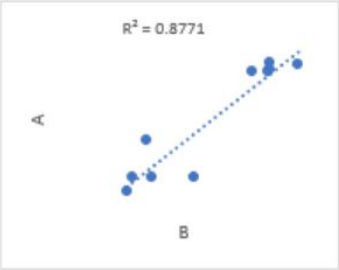
A	B	C
0.93	1.40	1.53
0.24	0.46	0.76
0.24	0.80	1.53
0.95	1.24	1.46
0.23	0.57	1.28
0.90	0.46	1.28
0.15	0.42	1.53
0.47	0.54	0.63
0.89	1.14	1.28
0.89	1.23	1.45

A	B	C
0.93	1.40	1.53
0.24	0.46	0.76
0.24	0.80	1.53
0.95	1.24	1.46
0.23	0.57	1.28
0.90	1.24	1.28
0.15	0.42	1.53
0.47	0.54	0.63
0.89	1.14	1.28
0.89	1.23	1.45









In this example, features A and B are known to be strongly correlated. See correlation including imputed values improve over updates

[1]van Burren, 2011. Figure credit: https://cran.r-project.org/web/packages/miceRanger/vignettes/miceAlgorithm.html

Example of imputation through prediction in the widely used MICE Algorithm[1]

A	B	C
0.93	1.40	1.53
0.24	0.46	0.76
	0.80	
0.95	1.24	1.46
0.23	0.57	
0.90		1.28
0.15	0.42	
0.47	0.54	0.63
	1.14	
0.89	1.23	1.45

A	B	C
0.93	1.40	1.53
0.24	0.46	0.76
0.90	0.80	1.53
0.95	1.24	1.46
0.23	0.57	1.28
0.90	0.46	1.28
0.15	0.42	1.53
0.47	0.54	0.63
0.47	1.14	1.28
0.89	1.23	1.45

A	B	C
0.93	1.40	1.53
0.24	0.46	0.76
0.24	0.80	1.53
0.95	1.24	1.46
0.23	0.57	1.28
0.90	0.46	1.28
0.15	0.42	1.53
0.47	0.54	0.63
0.89	1.14	1.28
0.89	1.23	1.45

A	B	C
0.93	1.40	1.53
0.24	0.46	0.76
0.24	0.80	1.53
0.95	1.24	1.46
0.23	0.57	1.28
0.90	1.24	1.28
0.15	0.42	1.53
0.47	0.54	0.63
0.89	1.14	1.28
0.89	1.23	1.45

Continue this update process for feature C, and then circle back to feature A and repeat process in cycles until imputed values for all features have converged

[1]van Burren, 2011. Figure credit: https://cran.r-project.org/web/packages/miceRanger/vignettes/miceAlgorithm.html

Example of imputation through prediction in the widely used MICE Algorithm[1]

A	B	C
0.93	1.40	1.53
0.24	0.46	0.76
	0.80	
0.95	1.24	1.46
0.23	0.57	
0.90		1.28
0.15	0.42	
0.47	0.54	0.63
	1.14	
0.89	1.23	1.45

A	B	C
0.93	1.40	1.53
0.24	0.46	0.76
0.90	0.80	1.53
0.95	1.24	1.46
0.23	0.57	1.28
0.90	0.46	1.28
0.15	0.42	1.53
0.47	0.54	0.63
0.47	1.14	1.28
0.89	1.23	1.45

A	B	C
0.93	1.40	1.53
0.24	0.46	0.76
0.24	0.80	1.53
0.95	1.24	1.46
0.23	0.57	1.28
0.90	0.46	1.28
0.15	0.42	1.53
0.47	0.54	0.63
0.89	1.14	1.28
0.89	1.23	1.45

A	B	C
0.93	1.40	1.53
0.24	0.46	0.76
0.24	0.80	1.53
0.95	1.24	1.46
0.23	0.57	1.28
0.90	1.24	1.28
0.15	0.42	1.53
0.47	0.54	0.63
0.89	1.14	1.28
0.89	1.23	1.45

Continue this update process for feature C, and then circle back to feature A and repeat process in cycles until imputed values for all features have converged

Full MICE Algorithm (multiple imputation) repeats this for N random initializations of the dataset and then aggregates for final imputation + uncertainty measure. We will not cover different initialization methods and implications.

[1]van Burren, 2011. Figure credit: https://cran.r-project.org/web/packages/miceRanger/vignettes/miceAlgorithm.html

Sources of EHR data

- Open-source EHR datasets (MIMIC-III/IV, MIMIC-CXR, …)
- Restricted EHR data from individual institutions
  - Major vendors: EPIC, Cerner, etc.
- Also: insurance claims data
  - Fills in blanks of patient health outside the hospital!
    - Visits with other care providers outside the hospital EHR system
    - Pharmacy visits

Sources of EHR data

- Open-source EHR datasets (MIMIC-III/IV, MIMIC-CXR, …)
- Restricted EHR data from individual institutions
  - Major vendors: EPIC, Cerner, etc.
- Also: insurance claims data
  - Fills in blanks of patient health outside the hospital!
    - Visits with other care providers outside the hospital EHR system
    - Pharmacy visits

Challenge: many of these data sources are in their own formats. How do we use multiple data sources?

OMOP Common Data Model

- Observational Medical Outcomes Partnership (OMOP)
- Created from public-private partnership involving FDA, pharmaceutical companies, and healthcare providers
- Standardized format and vocabulary
- Allows conversion of patient data from different sources into a common structure for analysis
- Intended to support data analysis

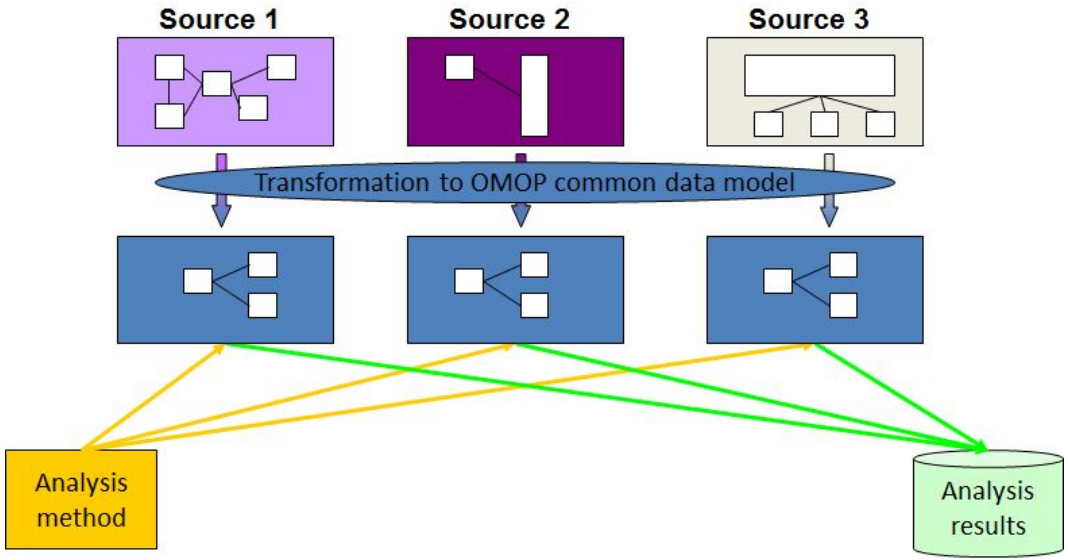


Figure credit: https://www.ohdsi.org/wp-content/uploads/2014/07/Why-CDM.png

OMOP Common Data Model

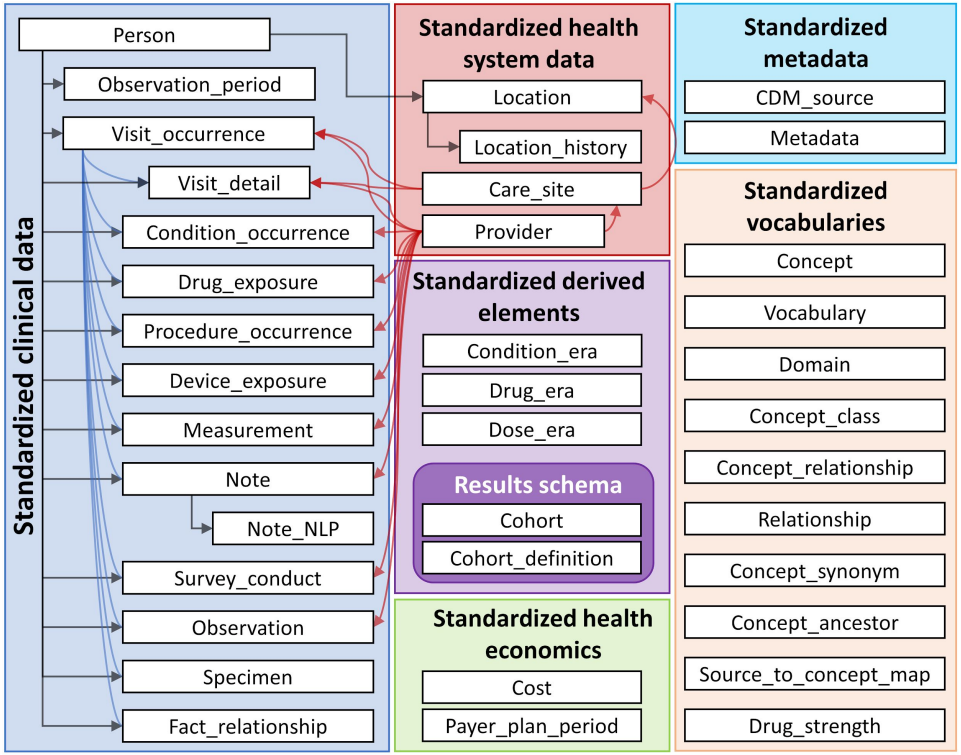


Figure credit: https://ohdsi.github.io/TheBookOfOhdsi/images/CommonDataModel/cdmDiagram.png

STARR: Stanford Hospital Data in OMOP

Stanford MEDICINE | Observational Medical Outcomes Partnership
STAnford Research data Repository

SUMMARY ACCESS LEARN NERO

Stanford Electronic Health Records in OMOP

STARR-OMOP is Stanford Electronic Health Record data from its two Hospitals in a Observational Medical Outcomes Partnership (OMOP) Common Data Model (CDM). Use OMOP for observational science, population health science, collaborative network studies and reproducible data science.

Standardized Data

- Standardized vocabulary
- Transparent data transformations
- High mapping rate

FHIR

- Fast healthcare interoperability resources (FHIR)
- Web-based standards / framework for secure exchange of electronic healthcare information across disparate sources
- Based on “resource” elements that contain information to be exchanged, as a JSON or XML object

```
<Patient xmlns="http://hl7.org/fhir">
  <extension url="http://www.goodhealth.org/consent#trials">
    <valueCode value="renal"/>
  </extension>
  <text>
    <status value="generated"/>
    <div xmlns="http://www.w3.org/1999/xhtml">
      <p>Henry Levin the 7th</p>
      <p>MRN: 123456</p>
    </div>
  </text>
  <identifier>
    <use value="usual"/>
    <label value="MRN"/>
    <system value="http://www.goodhealth.org/identifiers/mrn"/>
    <value value="123456"/>
  </identifier>
  <name>
    <family value="Levin"/>
    <given value="Henry"/>
    <suffix value="The 7th"/>
  </name>
  <gender>
    <text value="Male"/>
  </gender>
  <birthDate value="1932-09-24"/>
  <managingOrganization>
    <reference value="Organization/2"/>
    <display value="Good Health Clinic"/>
  </managingOrganization>
  <active value="true"/>
</Patient>
```

Extension with URL to definition

Human Readable Summary

Standard Data:
- MRN
- Name
- Gender
- Birth Date
- Provider

Figure credit: https://www.hl7.org/fhir/DSTU1/shot.png

FHIR

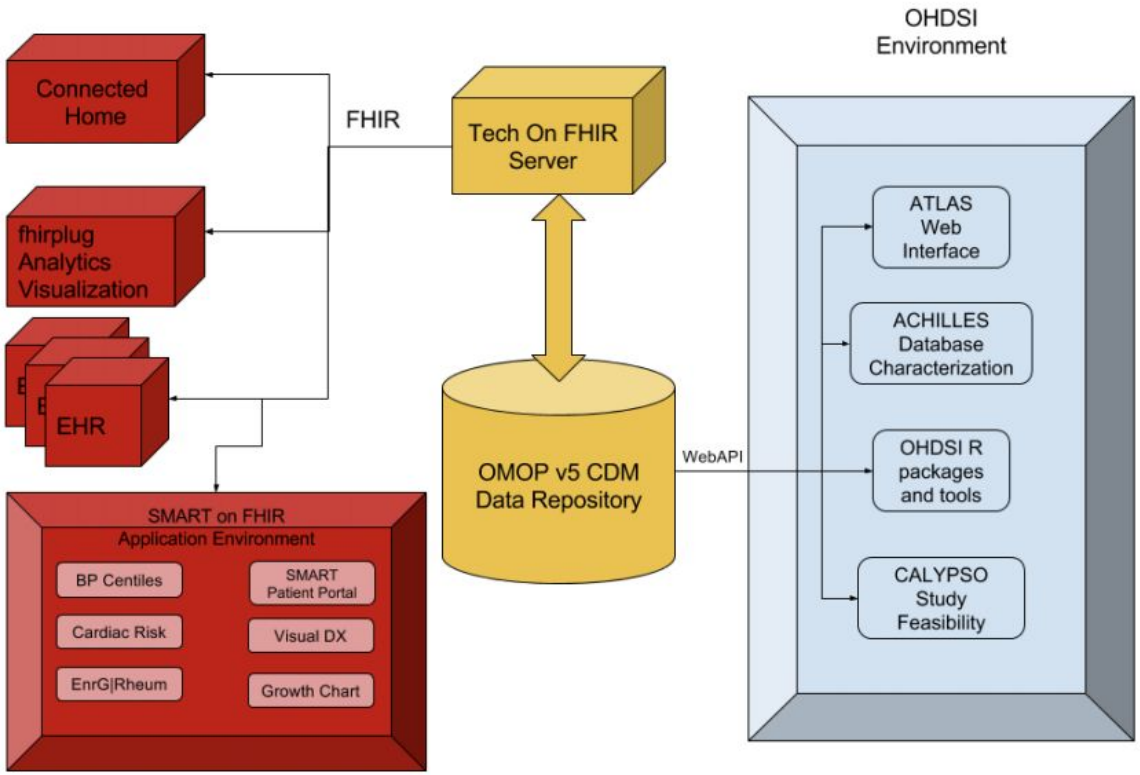


Figure credit: Choi et al. OHDSI on FHIR Platform Development with OMOP CDM mapping to FHIR Resources. 2016.

FHIR

FHIR-based information exchange between different sources

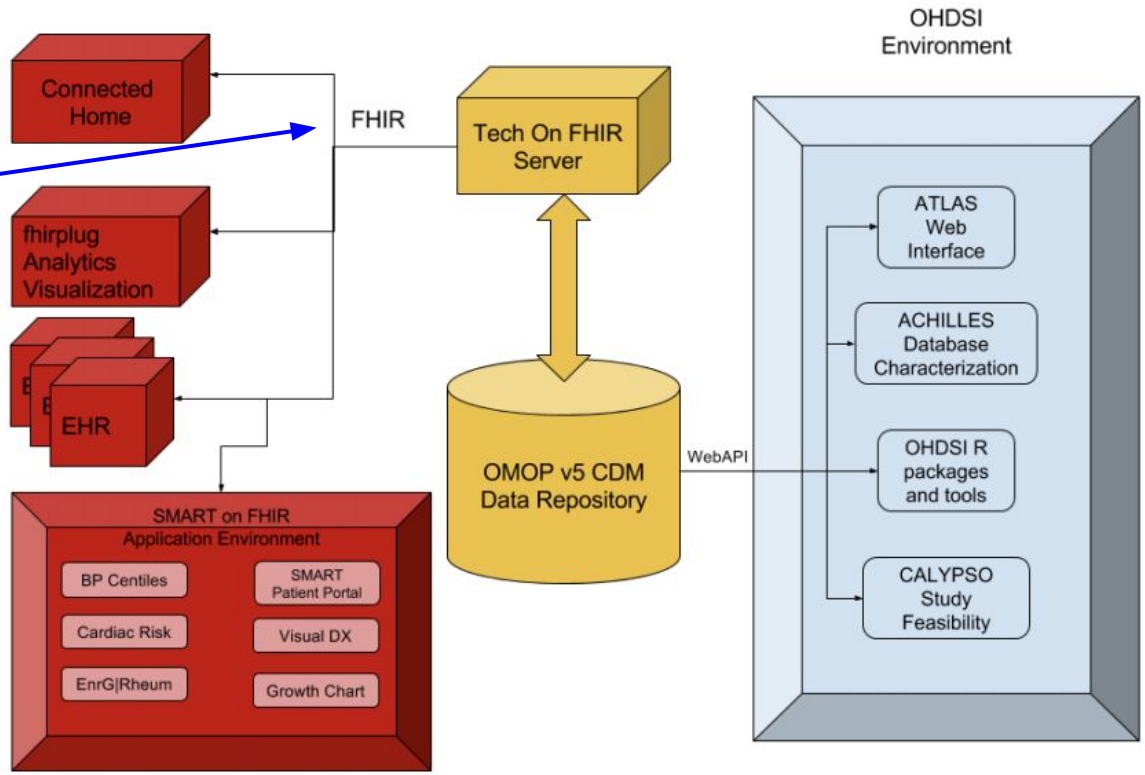


Figure credit: Choi et al. OHDSI on FHIR Platform Development with OMOP CDM mapping to FHIR Resources. 2016.

FHIR

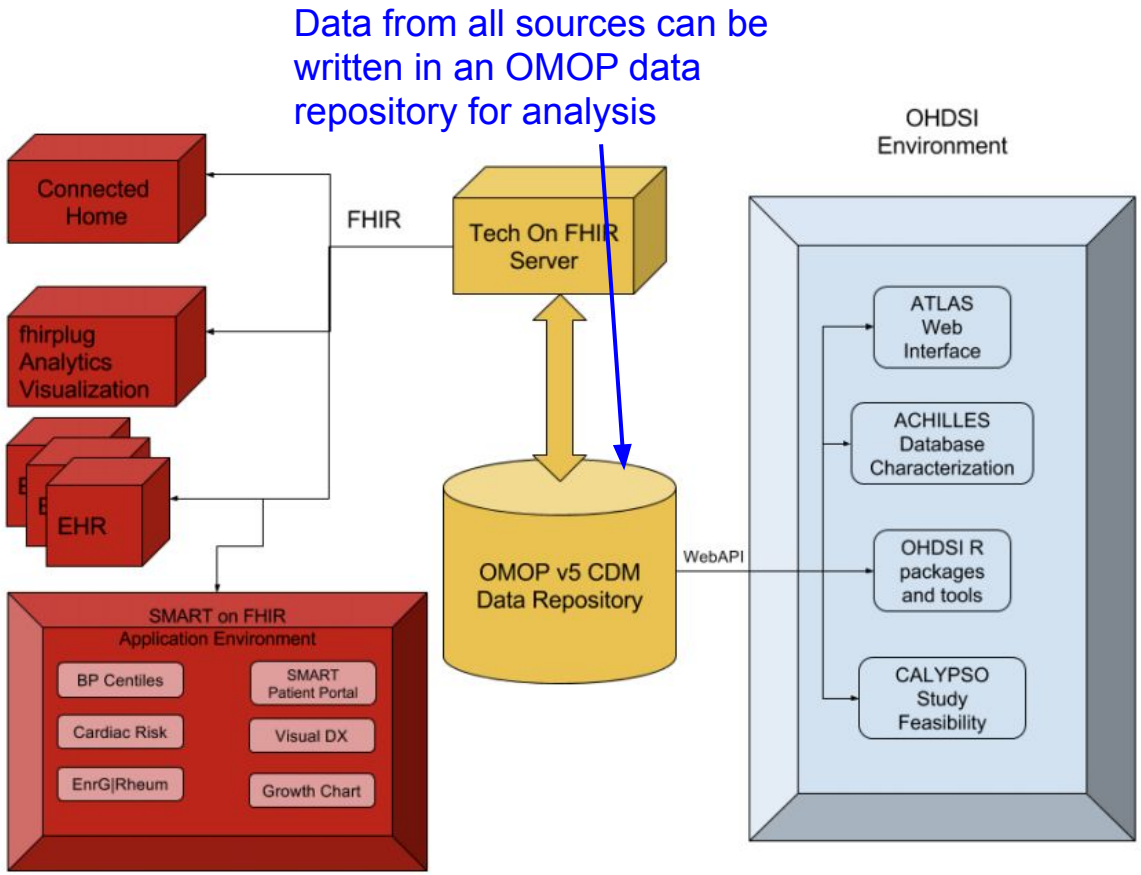


Figure credit: Choi et al. OHDSI on FHIR Platform Development with OMOP CDM mapping to FHIR Resources. 2016.

FHIR

OHDSI (parent of OMOP) also provides tools and resources for data analysis

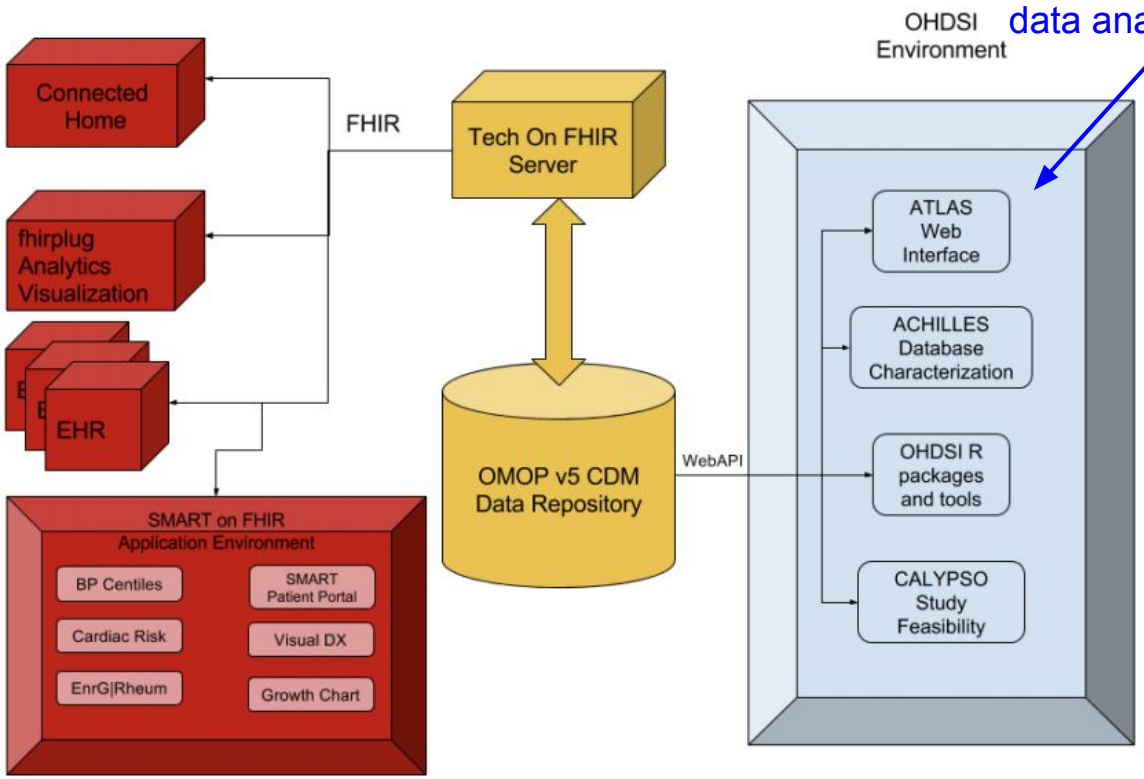
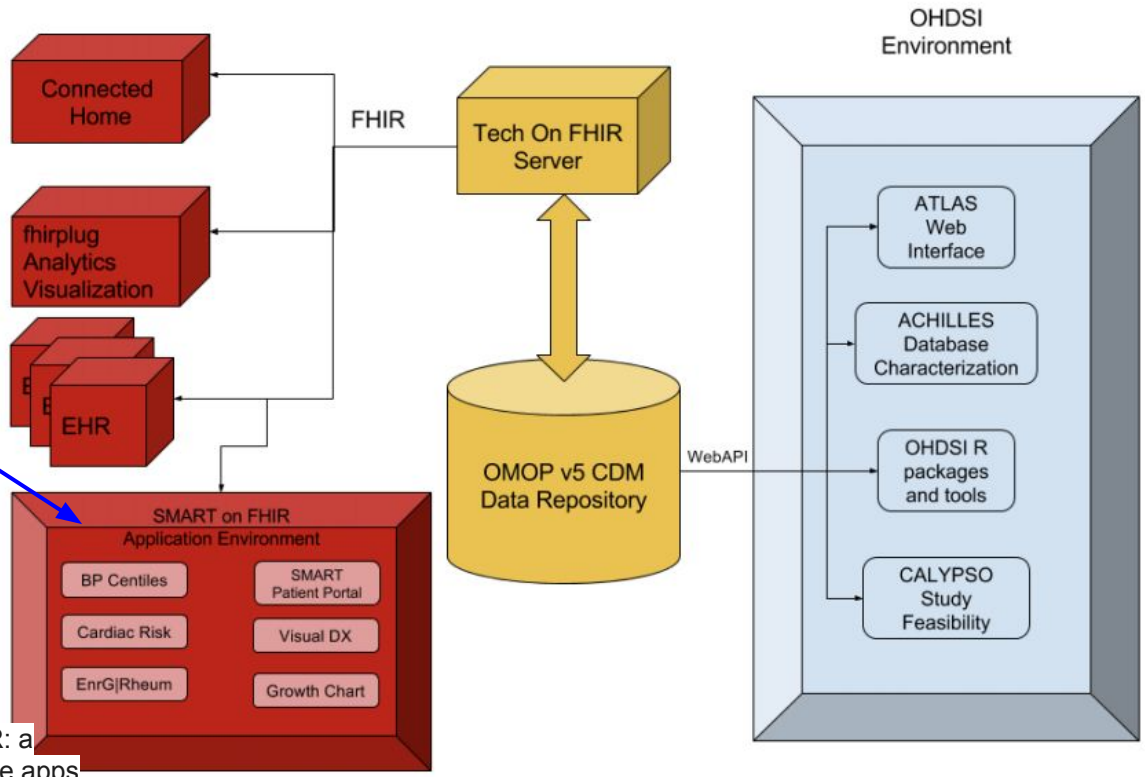


Figure credit: Choi et al. OHDSI on FHIR Platform Development with OMOP CDM mapping to FHIR Resources. 2016.

FHIR

SMART on FHIR is a platform for building third-party apps that interface with health data in e.g. EHRs, through FHIR.



Mandel et al. SMART on FHIR: a standards-based, interoperable apps platform for electronic health records. JAMA, 2016.

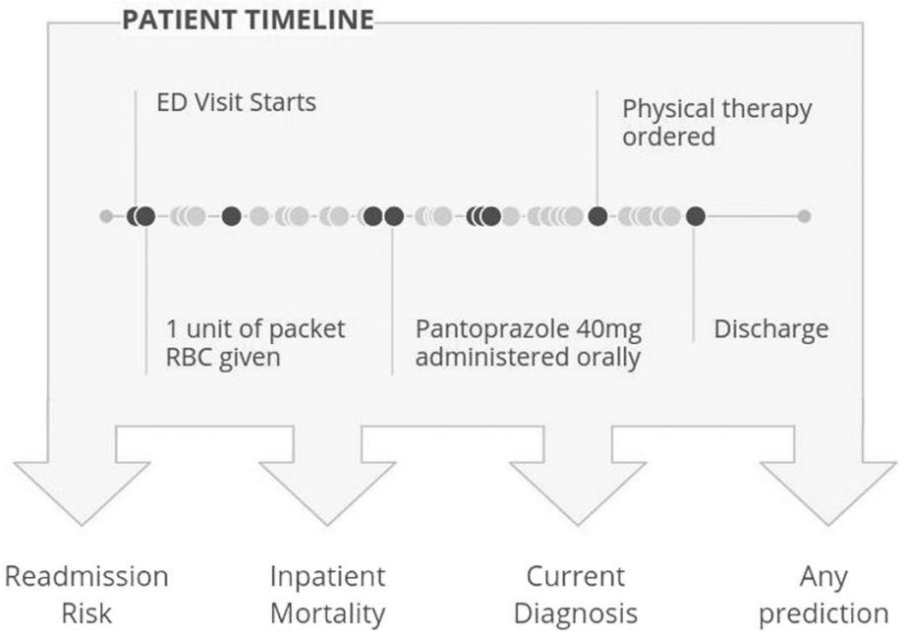
Figure credit: Choi et al. OHDSI on FHIR Platform Development with OMOP CDM mapping to FHIR Resources. 2016.

Aside: improving EHR technology and utility major current issue in healthcare

- **Have already seen one challenge: interoperability**
  - EHR systems were built and adopted very quickly -- not enough time to design for interoperability
- **Are EHRs being used meaningfully?**
  - Clinicians spending huge amount of time on documentation and interfacing with EHR system -> burnout and reduced patient interaction
  - Lots of pain points. What are the benefits?
- **Ongoing efforts to reduce pain points**
  - Improving user experience and AI-assisted documentation (dictation, autocomplete, etc.)
- **Ongoing efforts to improve value**
  - Data analytics, clinical decision support

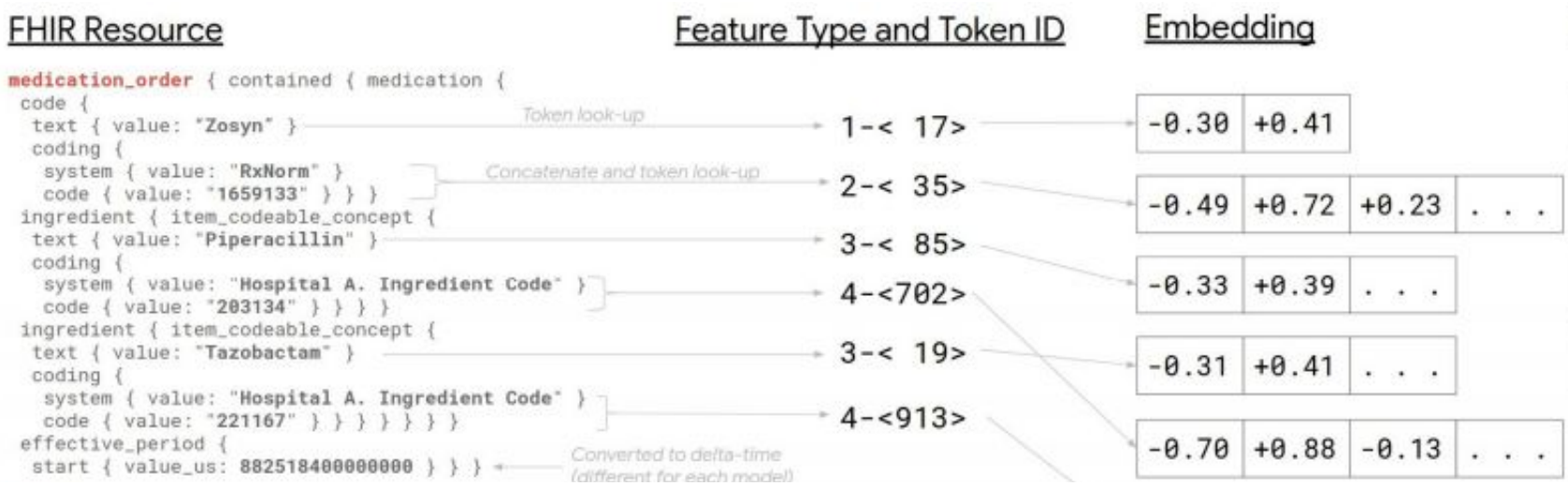
Rajkomar et al. 2018

- Clinical predictions from patients' entire raw EHR records, in FHIR format
- De-identified EHR data from two US academic centers with 216,221 adult patients
- Prediction tasks: in-hospital mortality, 30-day unplanned readmission, prolonged length of stay, patients' final discharge diagnoses
- 46,864,534,945 total data points across data (every event, every word in note, etc.)



Rajkomar et al. Scalable and accurate deep learning with electronic health records. Npj Digital Medicine, 2018.

Data representation

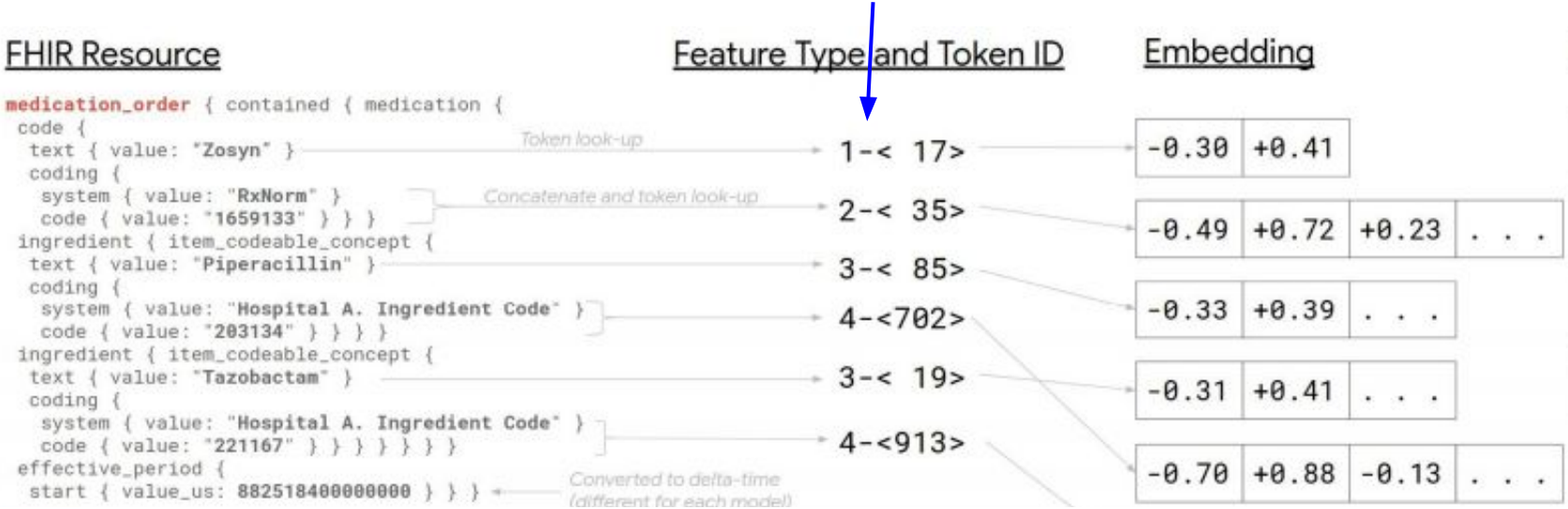


Raw data as FHIR resources

Rajkomar et al. Scalable and accurate deep learning with electronic health records. Npj Digital Medicine, 2018.

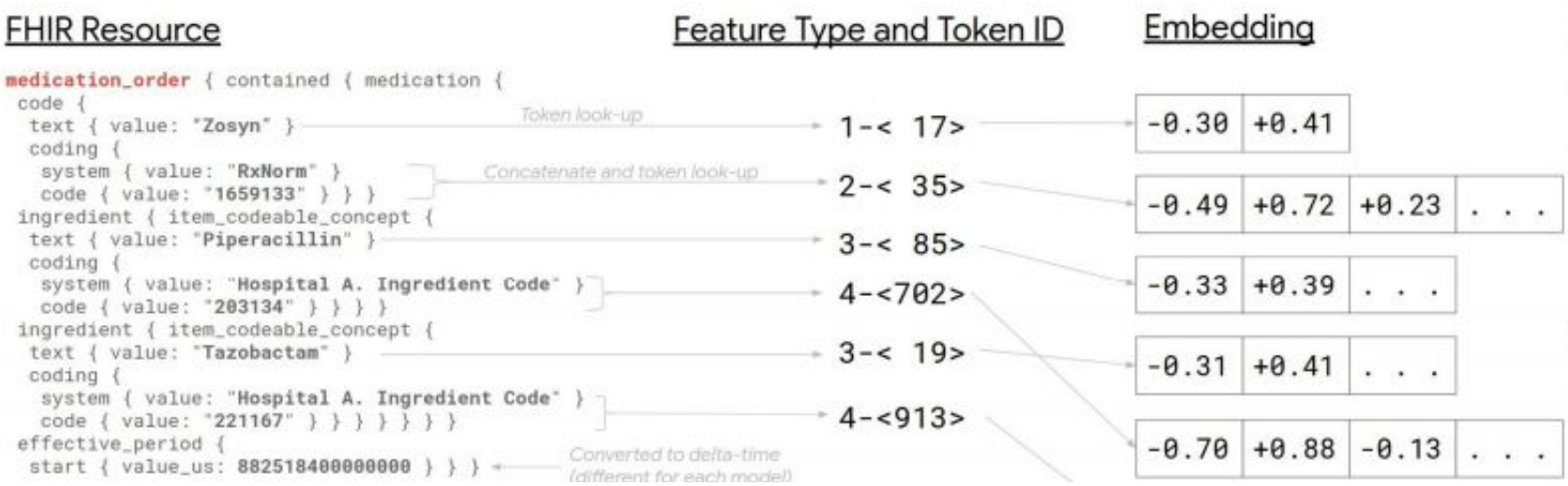
Data representation

Each element is mapped to a token ID (e.g. medication=zosyn), with a token "feature type"



Rajkomar et al. Scalable and accurate deep learning with electronic health records. Npj Digital Medicine, 2018.

Data representation



Every unique token is numerically represented by an “embedding vector” that will represent the token in the model. The embedding vector values are learned; similar tokens will probably have similar embedding vectors.

Rajkomar et al. Scalable and accurate deep learning with electronic health records. Npj Digital Medicine, 2018.

Token embeddings

[0 0 1 0 0 0 0 …. 0] X

1xN token input (one-hot selection of token)

0.5	0.2	0.1
0.6	0.1	0.6
0.5	0.8	0.2
0.7	0.9	0.3
0.3	0.5	0.1
…		
0.7	0.8	0.1

N x D embedding matrix

= [0.5 0.8 0.2]

D-dim token embedding

Token embeddings

[0 0 1 0 0 0 0 …. 0] X

1xN token input (one-hot selection of token)

0.5	0.2	0.1
0.6	0.1	0.6
0.5	0.8	0.2
0.7	0.9	0.3
0.3	0.5	0.1
…		
0.7	0.8	0.1

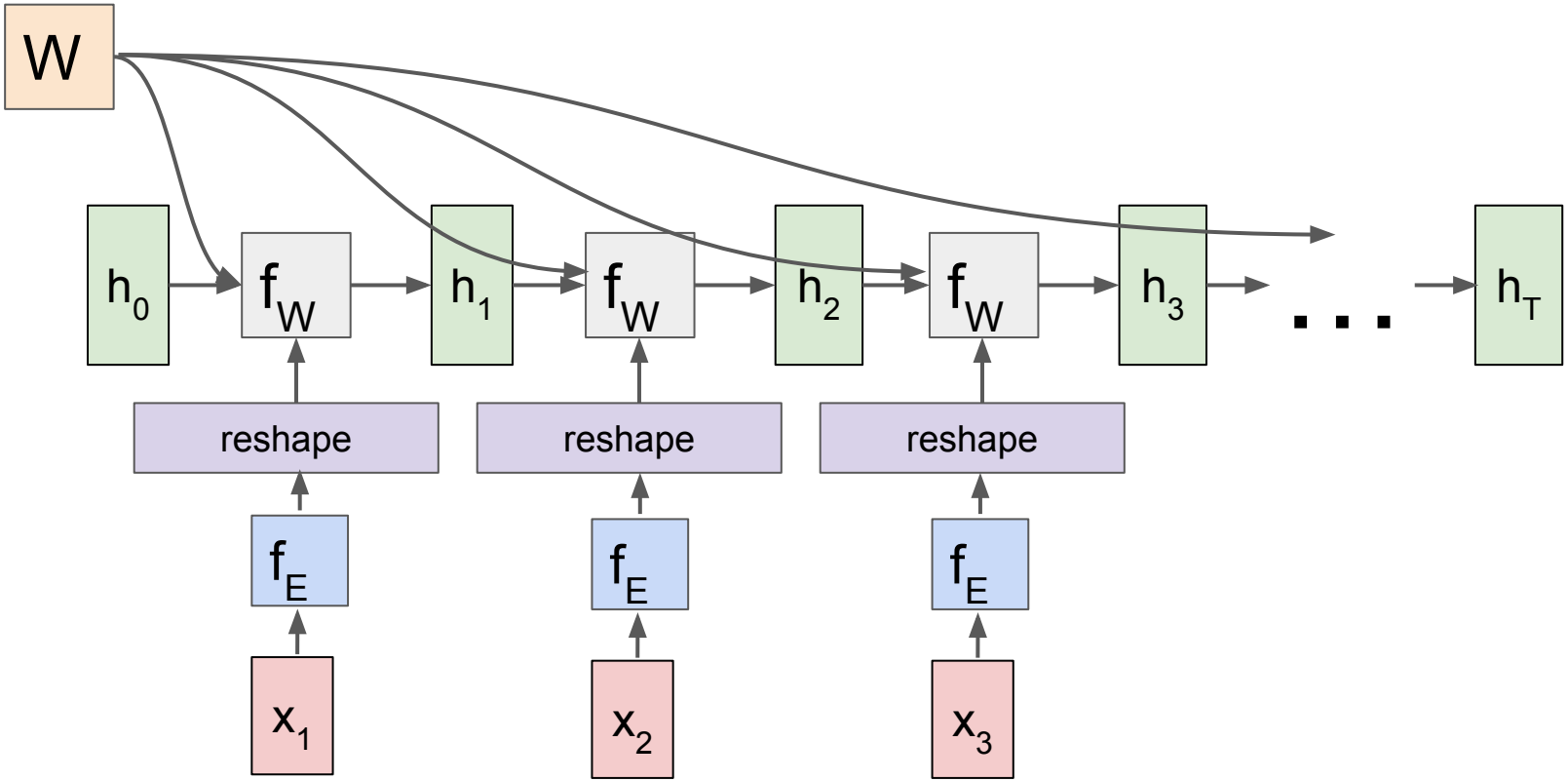
N x D embedding matrix

= [0.5 0.8 0.2]

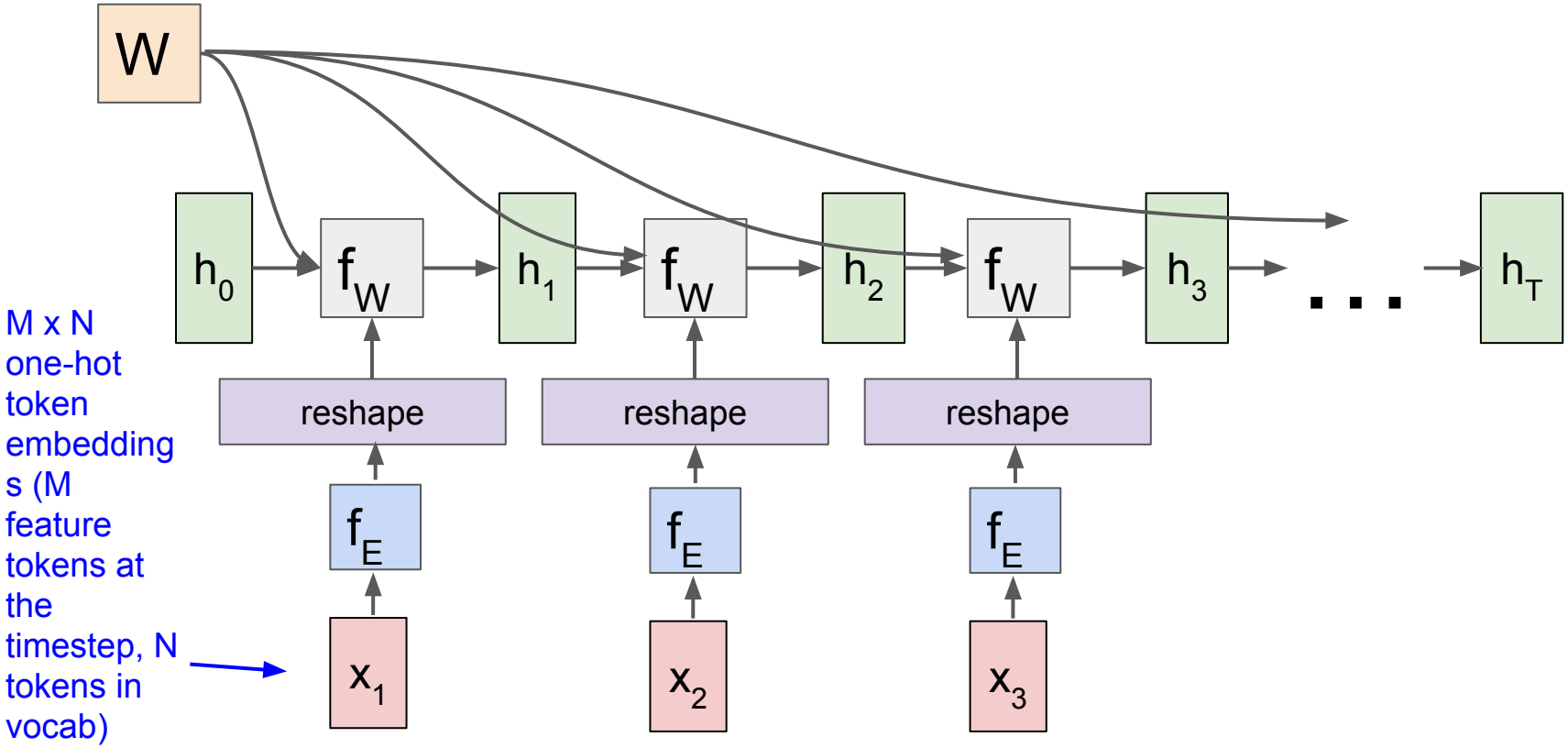
D-dim token embedding

In general, learning embedding matrices are a useful way to map discrete data into a semantically meaningful, continuous space! Will see frequently in natural language processing.

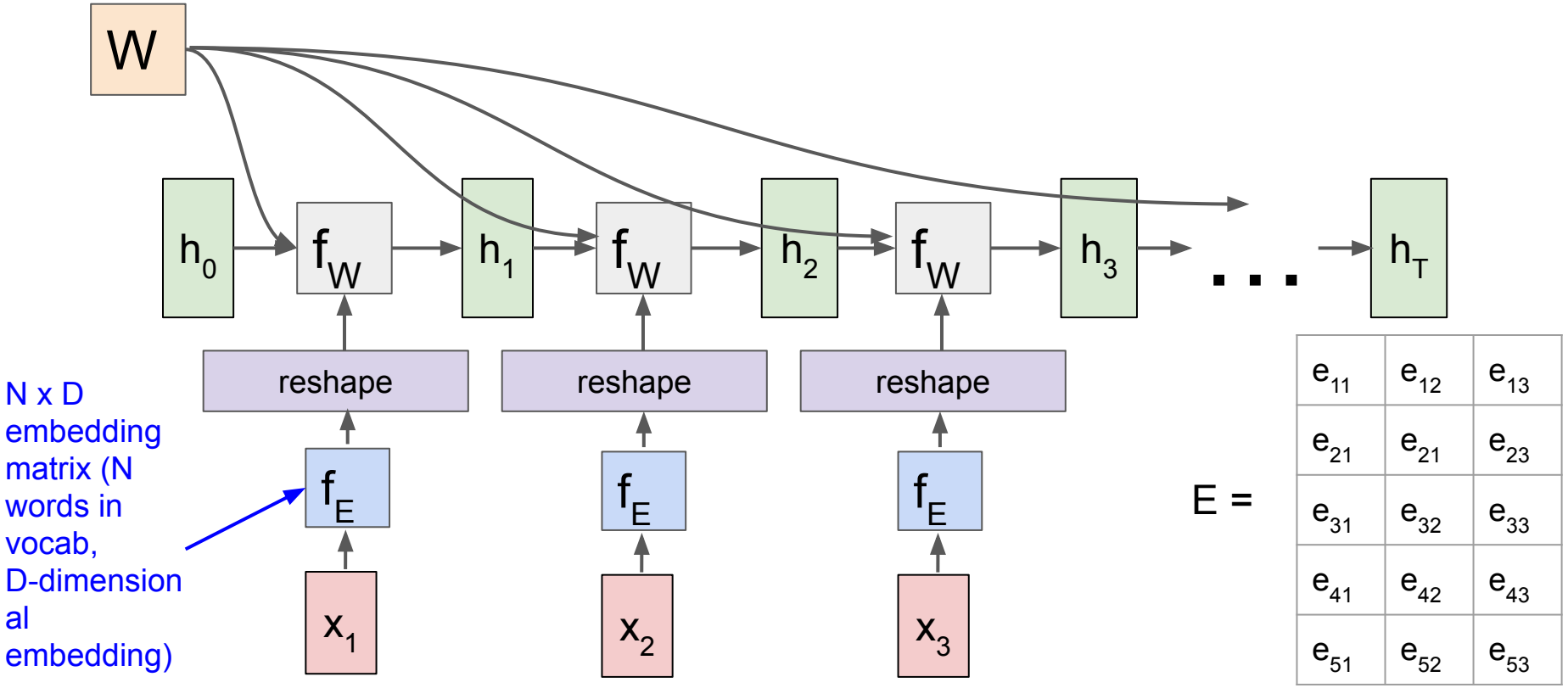
Computational graph input to RNN



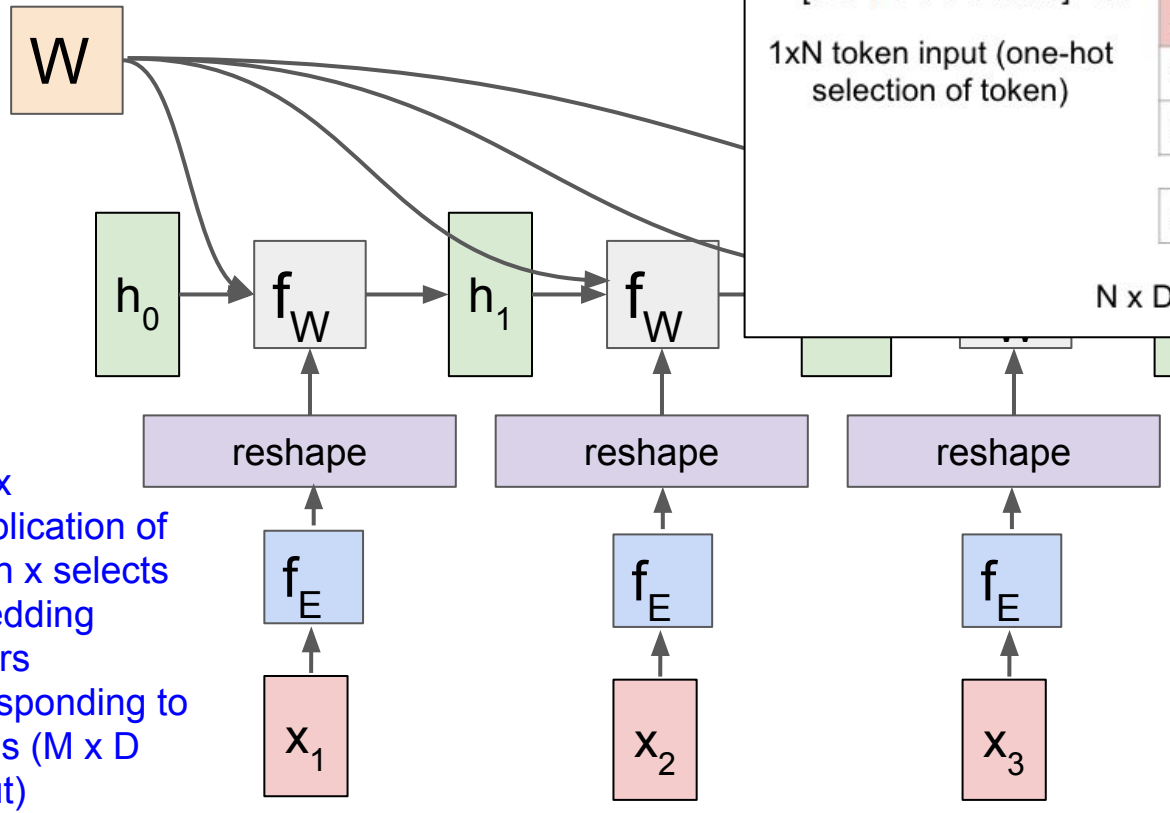
Computational graph input to RNN



Computational graph input to RNN



Computational graph inp



[0 0 1 0 0 0 0 …. 0] X

1xN token input (one-hot selection of token)

0.5	0.2	0.1
0.6	0.1	0.6
0.5	0.8	0.2
0.7	0.9	0.3
0.3	0.5	0.1
…		
0.7	0.8	0.1

N x D embedding matrix

= [0.5 0.8 0.2]

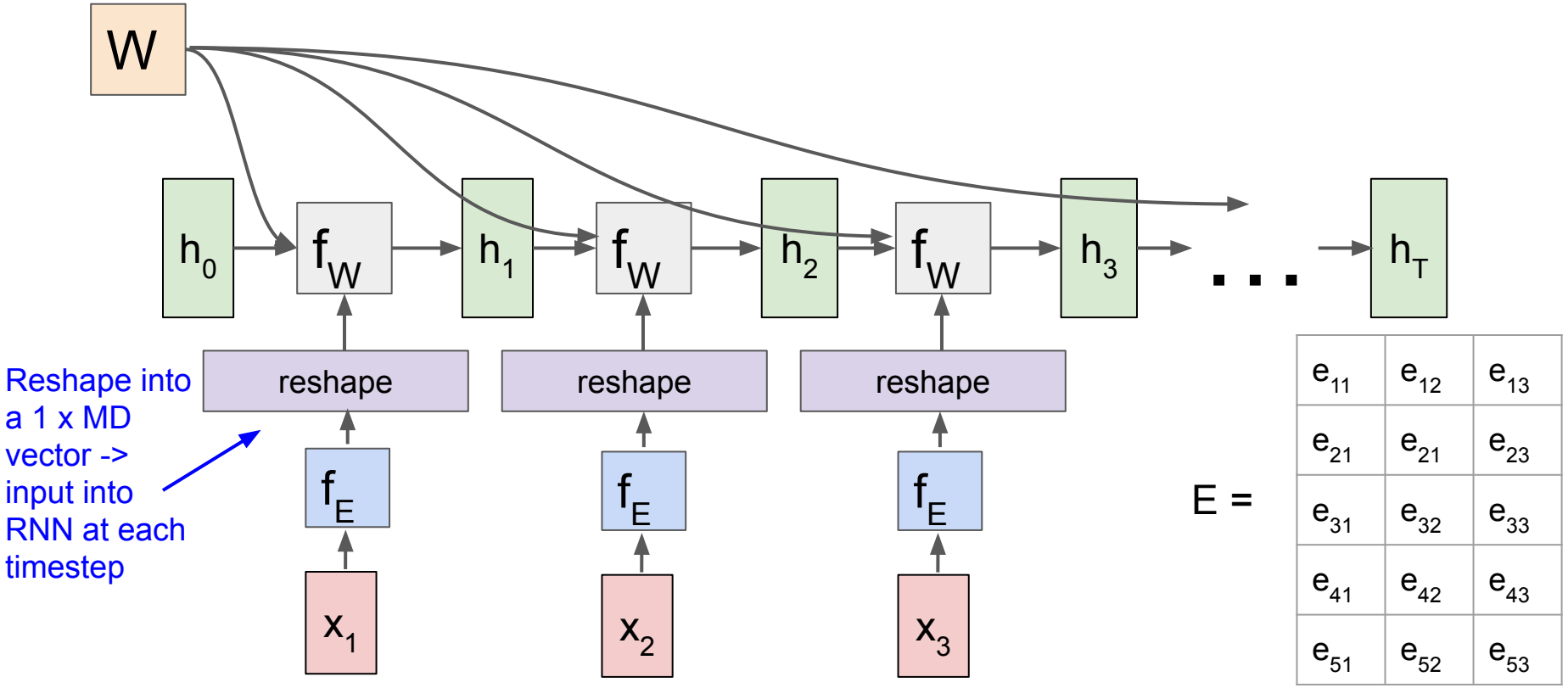
D-dim token embedding

Matrix multiplication of f_E with x selects embedding vectors corresponding to tokens (M x D output)

E =

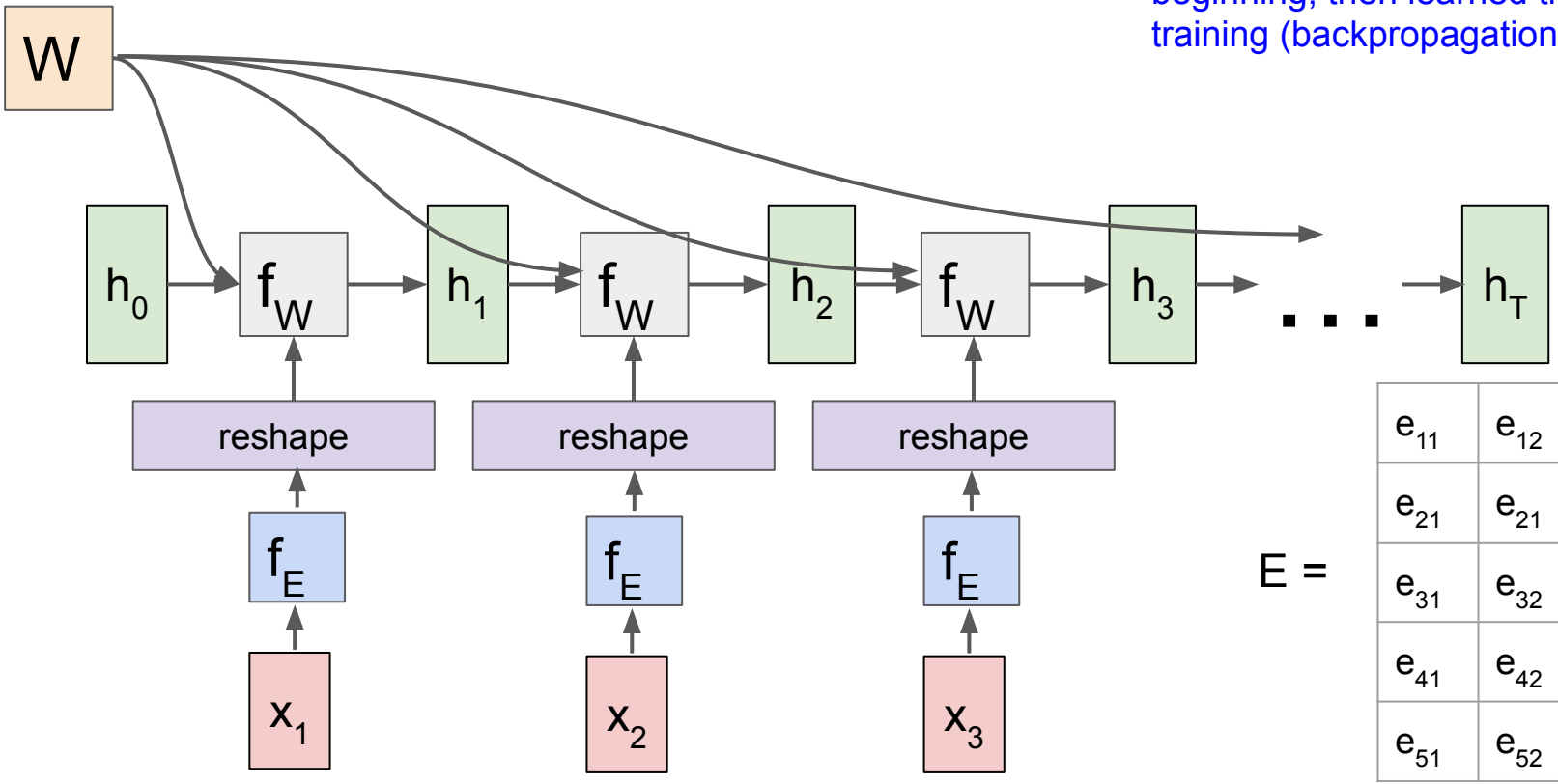
e_{11}	e_{12}	e_{13}
e_{21}	e_{21}	e_{23}
e_{31}	e_{32}	e_{33}
e_{41}	e_{42}	e_{43}
e_{51}	e_{52}	e_{53}

Computational graph input to RNN



Computational graph input to RNN

Embedding matrix has values that are randomly initialized at the beginning, then learned through training (backpropagation)!



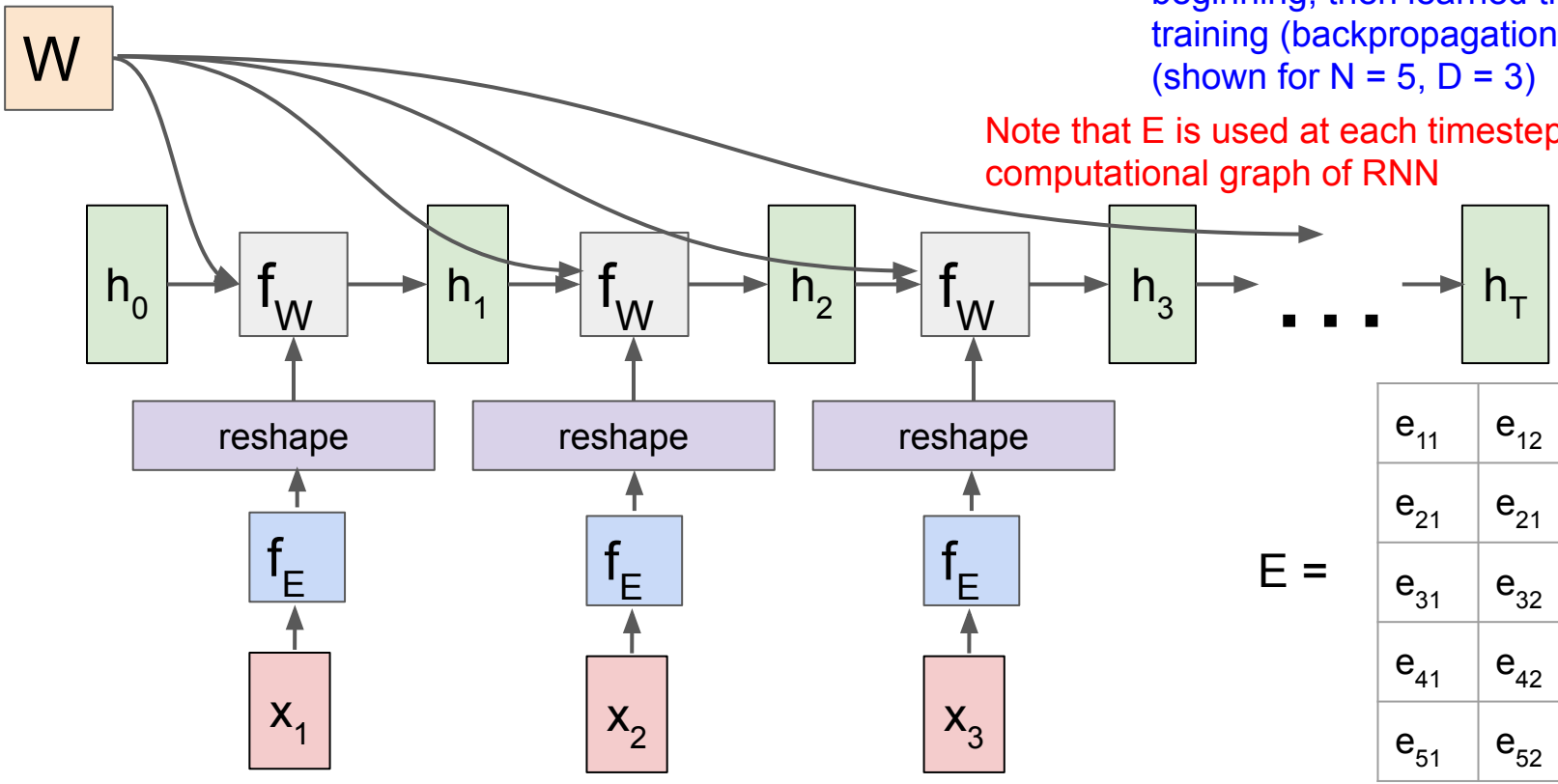
E =

e_{11}	e_{12}	e_{13}
e_{21}	e_{21}	e_{23}
e_{31}	e_{32}	e_{33}
e_{41}	e_{42}	e_{43}
e_{51}	e_{52}	e_{53}

Computational graph input to RNN

Embedding matrix has values that are randomly initialized at the beginning, then learned through training (backpropagation)! (shown for N = 5, D = 3)

Note that E is used at each timestep in computational graph of RNN



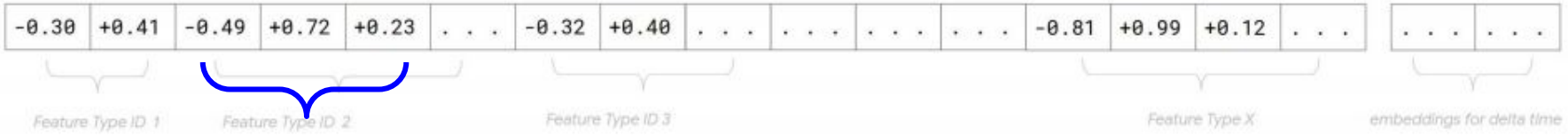
E =

e_{11}	e_{12}	e_{13}
e_{21}	e_{21}	e_{23}
e_{31}	e_{32}	e_{33}
e_{41}	e_{42}	e_{43}
e_{51}	e_{52}	e_{53}

Rajkomar et al. RNN (LSTM) input

Rajkomar et al. Scalable and accurate deep learning with electronic health records. Npj Digital Medicine, 2018.

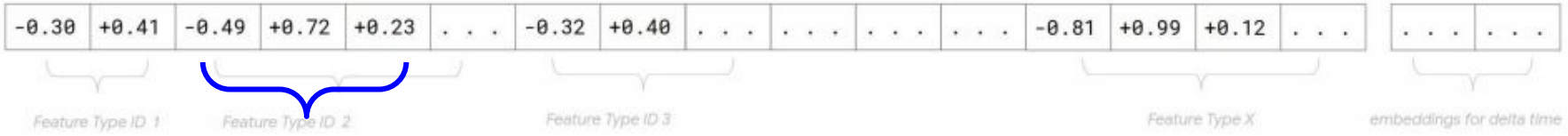
Rajkomar et al. RNN (LSTM) input



One vector representation for each token "feature type" (e.g. medication, procedure). Embeddings of multiple tokens corresponding to a same feature type are combined through averaging.

Rajkomar et al. Scalable and accurate deep learning with electronic health records. Npj Digital Medicine, 2018.

Rajkomar et al. RNN (LSTM) input



One vector representation for each token "feature type" (e.g. medication, procedure). Embeddings of multiple tokens corresponding to a same feature type are combined through averaging.

A little bit of added complexity: each feature type has its own embedding dimension D. A hyperparameter!

Rajkomar et al. Scalable and accurate deep learning with electronic health records. Npj Digital Medicine, 2018.

Rajkomar et al. RNN (LSTM) input

Also include an embedding representation of time delta from last RNN input.

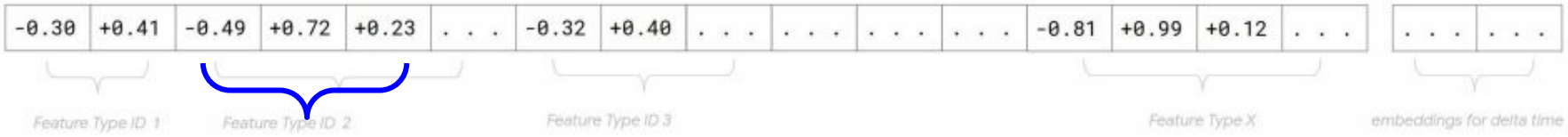
One vector representation for each token "feature type" (e.g. medication, procedure). Embeddings of multiple tokens corresponding to a same feature type are combined through averaging.

A little bit of added complexity: each feature type has its own embedding dimension D. A hyperparameter!

Rajkomar et al. Scalable and accurate deep learning with electronic health records. Npj Digital Medicine, 2018.

Rajkomar et al. RNN (LSTM) input

Also include an embedding representation of time delta from last RNN input.



One vector representation for each token "feature type" (e.g. medication, procedure). Embeddings of multiple tokens corresponding to a same feature type are combined through averaging.

A little bit of added complexity: each feature type has its own embedding dimension D. A hyperparameter!

Refer to paper for other details, e.g. bucketing of continuous data types into discrete token IDs.

Rajkomar et al. Scalable and accurate deep learning with electronic health records. Npj Digital Medicine, 2018.

Rajkomar et al.

Compared deep learning approach with baselines (e.g. logistic regression), and using all variables in data (flattened vector) vs hand-crafted features from subset of variables

	Hospital A	Hospital B
Inpatient Mortality, AUROC[1](95% CI)		
Deep learning 24 hours after admission	**0.95**(0.94-0.96)	**0.93**(0.92-0.94)
Full feature enhanced baseline at 24 hours after admission	0.93 (0.92-0.95)	0.91 (0.89-0.92)
Full feature simple baseline at 24 hours after admission	0.93 (0.91-0.94)	0.90 (0.88-0.92)
Baseline (aEWS[2]) at 24 hours after admission	0.85 (0.81-0.89)	0.86 (0.83-0.88)
30-day Readmission, AUROC (95% CI)		
Deep learning at discharge	**0.77**(0.75-0.78)	**0.76**(0.75-0.77)
Full feature enhanced baseline at discharge	0.75 (0.73-0.76)	0.75 (0.74-0.76)
Full feature simple baseline at discharge	0.74 (0.73-0.76)	0.73 (0.72-0.74)
Baseline (mHOSPITAL[3]) at discharge	0.70 (0.68-0.72)	0.68 (0.67-0.69)
Length of Stay at least 7 days AUROC (95% CI)		
Deep learning 24 hours after admission	**0.86**(0.86-0.87)	**0.85**(0.85-0.86)
Full feature enhanced baseline at 24 hours after admission	0.85 (0.84-0.85)	0.83 (0.83-0.84)
Full feature simple baseline at 24 hours after admission	0.83 (0.82-0.84)	0.81 (0.80-0.82)
Baseline (mLiu[4]) at 24 hours after admission	0.76 (0.75-0.77)	0.74 (0.73-0.75)

[1] Area under the receiver operator curve
[2] Augmented early warning score
[3] Modified HOSPITAL score
[4] Modified Liu score

Rajkomar et al. Scalable and accurate deep learning with electronic health records. Npj Digital Medicine, 2018.

Rajkomar et al.

Compared deep learning approach with baselines (e.g. logistic regression), and using all variables in data (flattened vector) vs hand-crafted features from subset of variables

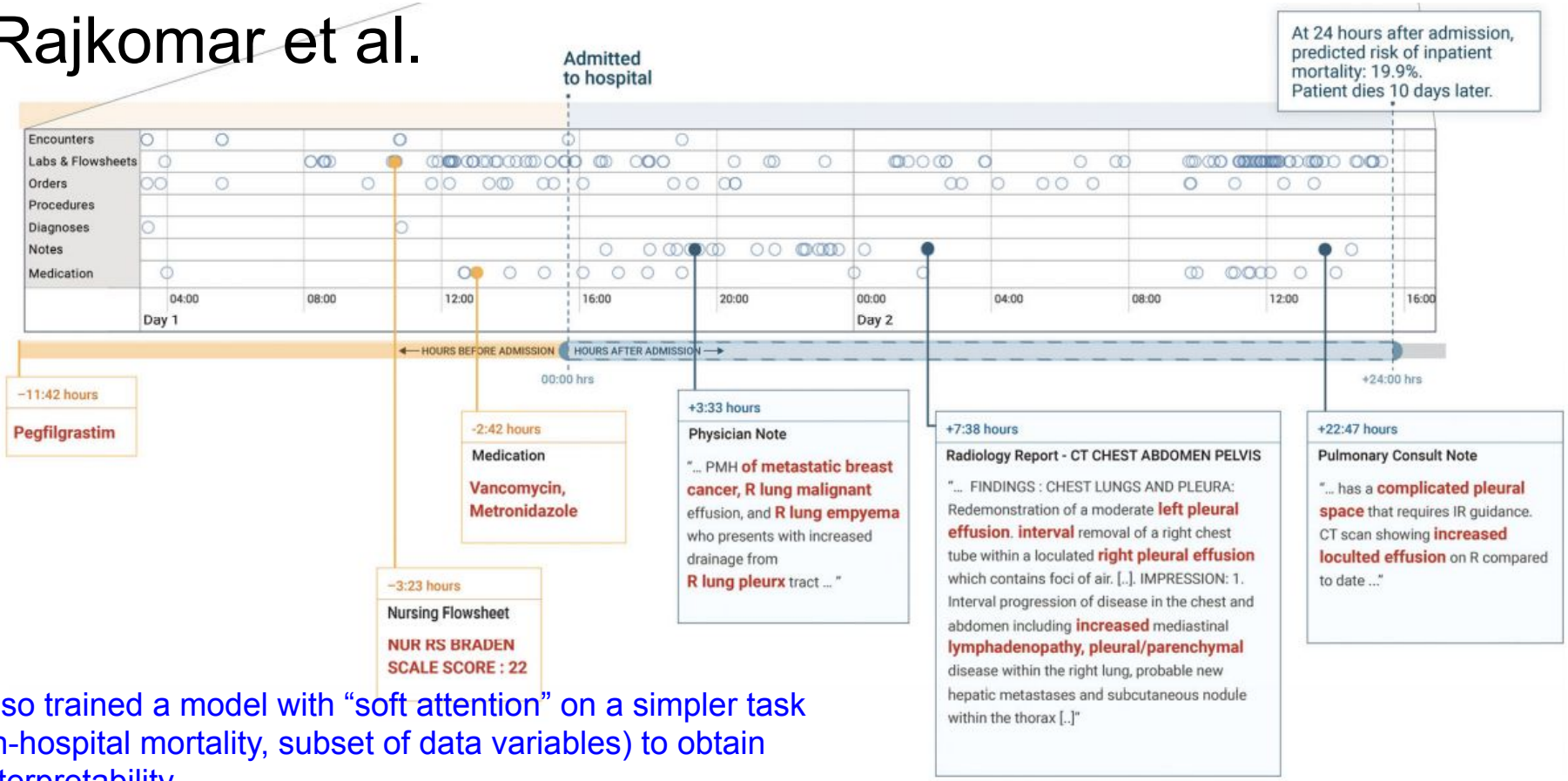
Evaluated model at different time points, e.g., at admission, 24 hrs after admission, discharge

	Hospital A	Hospital B
Inpatient Mortality, AUROC[1](95% CI)		
Deep learning 24 hours after admission	**0.95**(0.94-0.96)	**0.93**(0.92-0.94)
Full feature enhanced baseline at 24 hours after admission	0.93 (0.92-0.95)	0.91 (0.89-0.92)
Full feature simple baseline at 24 hours after admission	0.93 (0.91-0.94)	0.90 (0.88-0.92)
Baseline (aEWS[2]) at 24 hours after admission	0.85 (0.81-0.89)	0.86 (0.83-0.88)
30-day Readmission, AUROC (95% CI)		
Deep learning at discharge	**0.77**(0.75-0.78)	**0.76**(0.75-0.77)
Full feature enhanced baseline at discharge	0.75 (0.73-0.76)	0.75 (0.74-0.76)
Full feature simple baseline at discharge	0.74 (0.73-0.76)	0.73 (0.72-0.74)
Baseline (mHOSPITAL[3]) at discharge	0.70 (0.68-0.72)	0.68 (0.67-0.69)
Length of Stay at least 7 days AUROC (95% CI)		
Deep learning 24 hours after admission	**0.86**(0.86-0.87)	**0.85**(0.85-0.86)
Full feature enhanced baseline at 24 hours after admission	0.85 (0.84-0.85)	0.83 (0.83-0.84)
Full feature simple baseline at 24 hours after admission	0.83 (0.82-0.84)	0.81 (0.80-0.82)
Baseline (mLiu[4]) at 24 hours after admission	0.76 (0.75-0.77)	0.74 (0.73-0.75)

[1] Area under the receiver operator curve
[2] Augmented early warning score
[3] Modified HOSPITAL score
[4] Modified Liu score

Rajkomar et al. Scalable and accurate deep learning with electronic health records. Npj Digital Medicine, 2018.

Rajkomar et al.

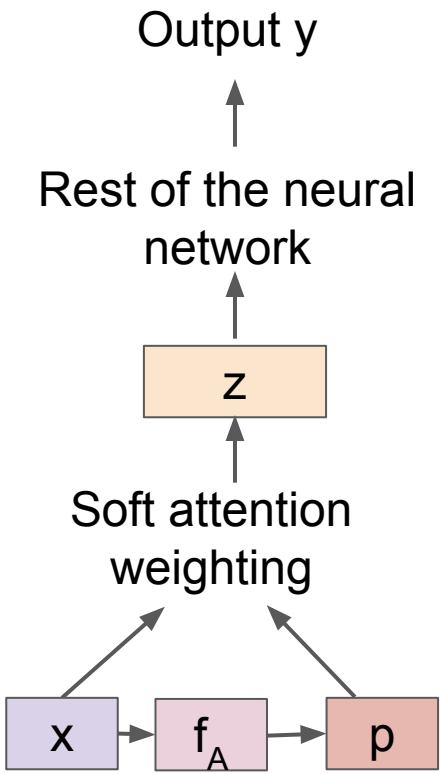


Also trained a model with "soft attention" on a simpler task (in-hospital mortality, subset of data variables) to obtain interpretability

Rajkomar et al. Scalable and accurate deep learning with electronic health records. Npj Digital Medicine, 2018.

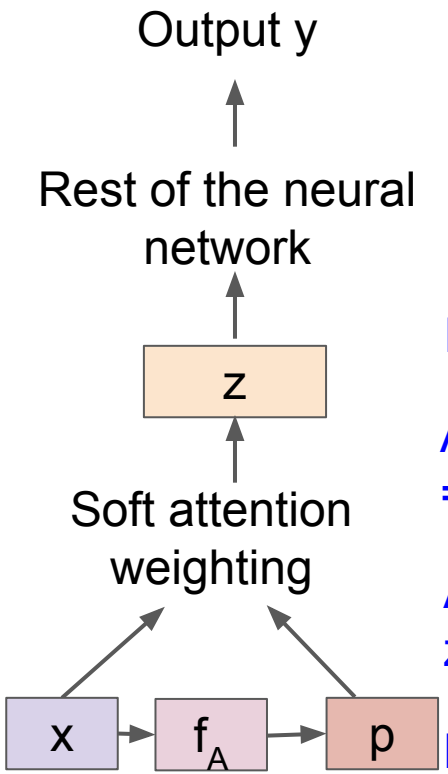
Soft attention

- Weight input variables by an "attention weights" vector p
- Learn to dynamically produce p for any given input, by making it a function of the input x and a fully connected layer f_A(with learnable parameters A)
- By optimizing for prediction performance, network will learn to produce p that gives stronger weights to the most informative features in x!



Soft attention

- Weight input variables by an "attention weights" vector p
- Learn to dynamically produce p for any given input, by making it a function of the input x and a fully connected layer f_A(with learnable parameters A)
- By optimizing for prediction performance, network will learn to produce p that gives stronger weights to the most informative features in x!



Input $x = [x_1, x_2, \ldots, x_D]$

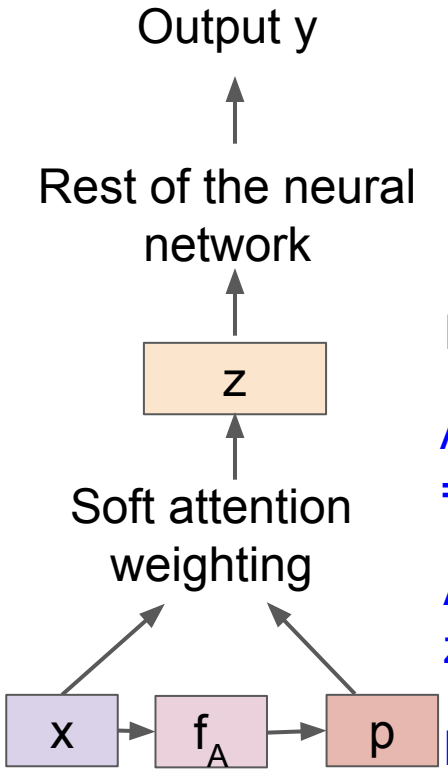
Attention weights p $=[p_1, p_2, \ldots, p_D]$

Attention-weighted input $z = [z_1, z_2, \ldots, z_D]$

Learnable fully connected layer f_A with weights A

Soft attention

- Weight input variables by an "attention weights" vector p
- Learn to dynamically produce p for any given input, by making it a function of the input x and a fully connected layer f_A(with learnable parameters A)
- By optimizing for prediction performance, network will learn to produce p that gives stronger weights to the most informative features in x!



p is output of a softmax function -> attention weights sum to 1

Input $x = [x_1, x_2, \ldots, x_D]$

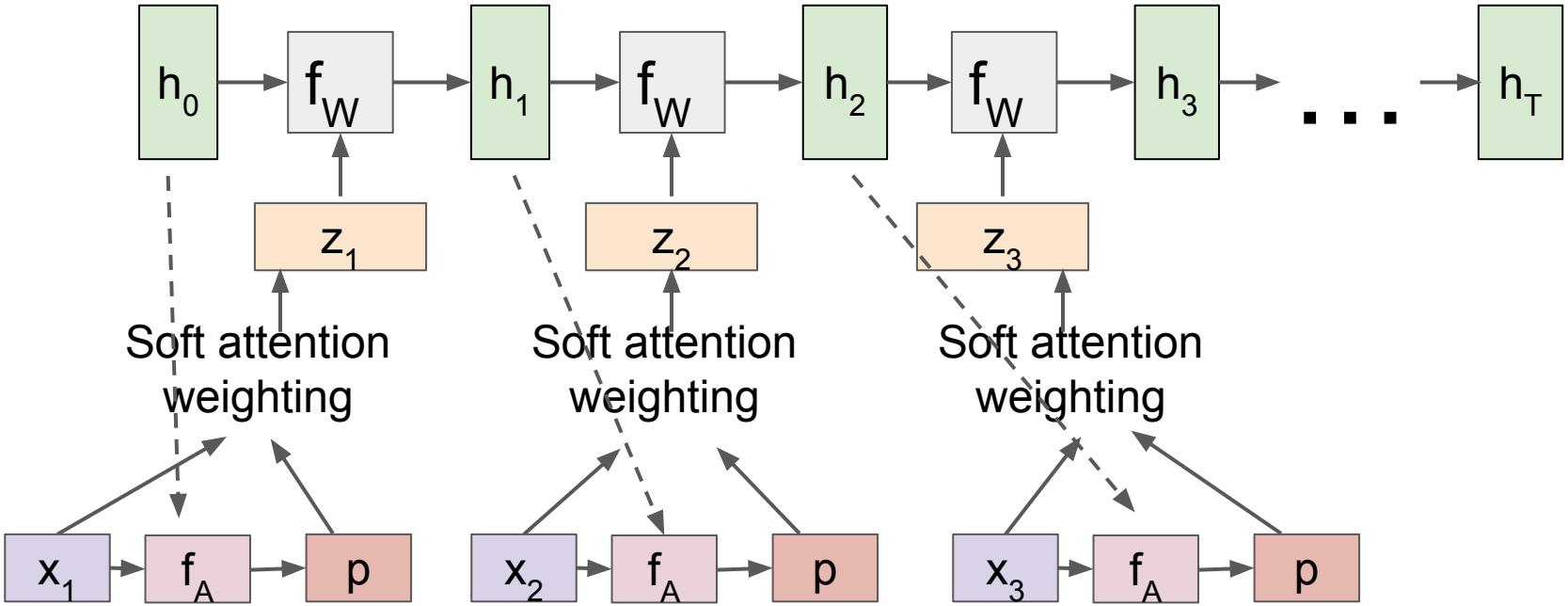
Attention weights p $=[p_1, p_2, \ldots, p_D]$

Attention-weighted input $z = [z_1, z_2, \ldots, z_D]$

Learnable fully connected layer f_A with weights A

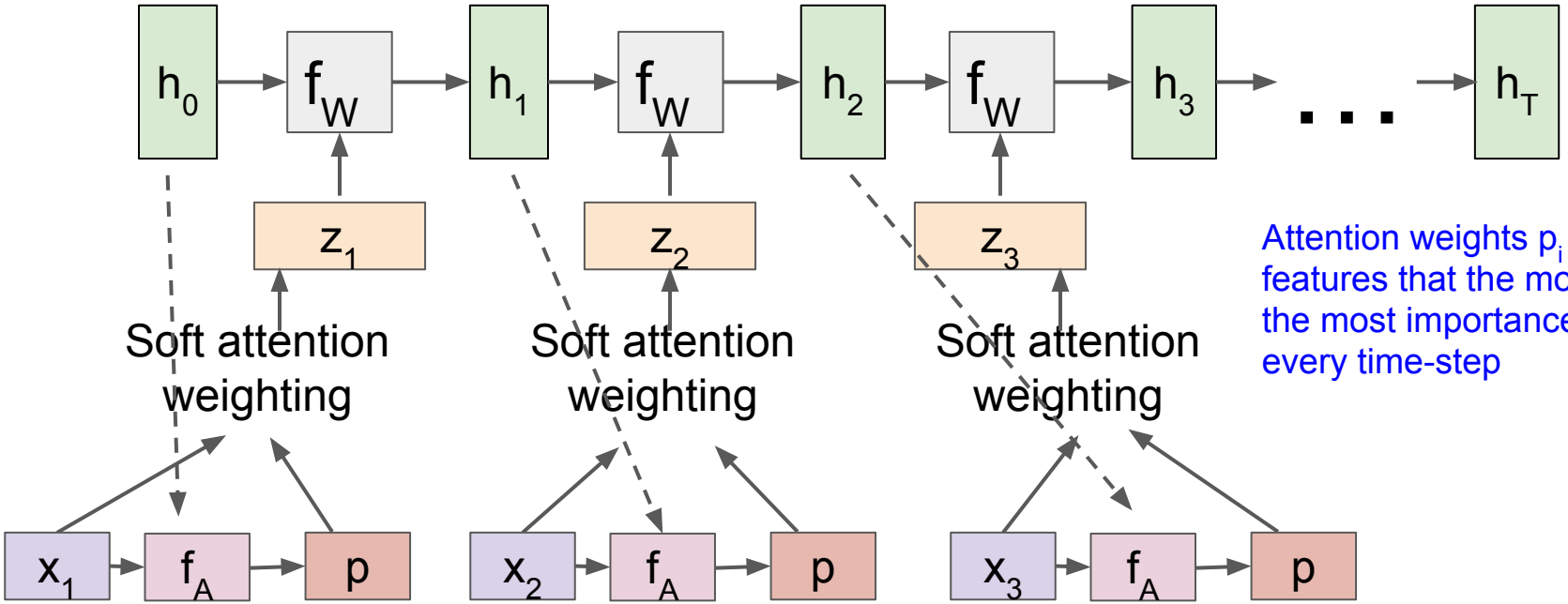
Soft attention in RNNs

Note that f_A produces attention weights as a function of both current input x as well as previous hidden state h!



Soft attention in RNNs

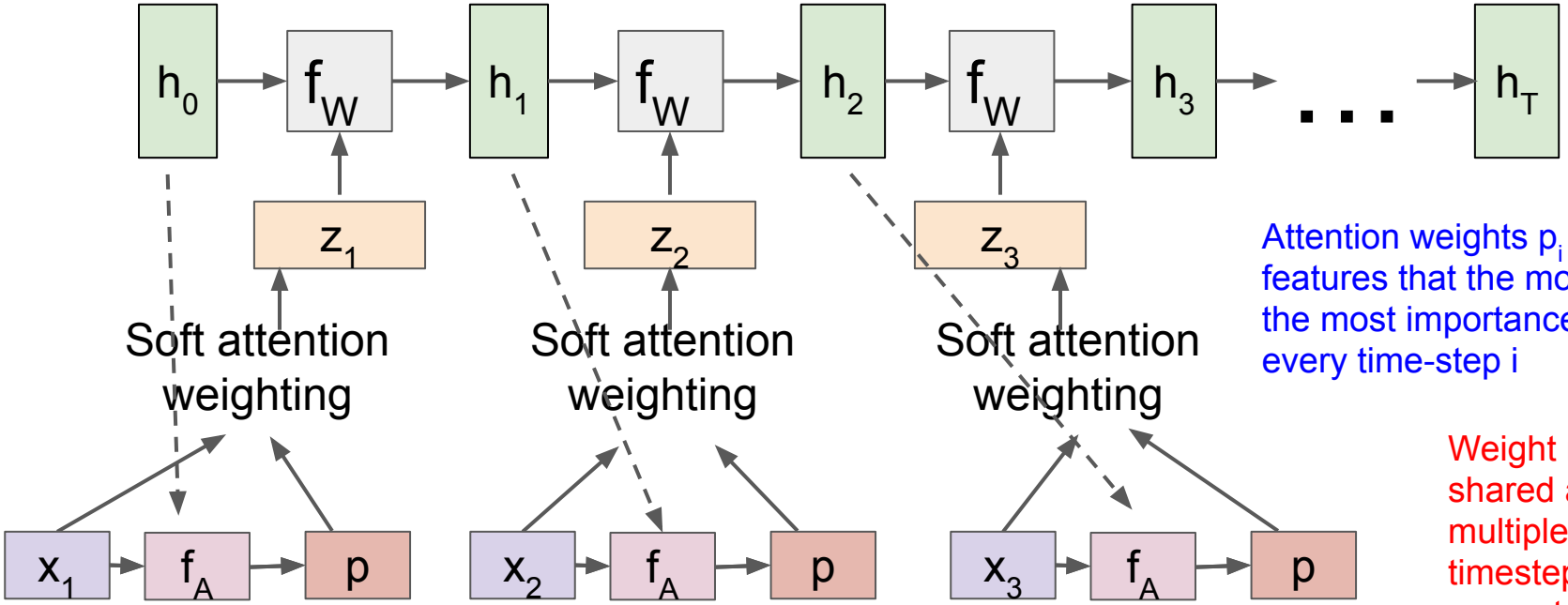
Note that f_A produces attention weights as a function of both current input x as well as previous hidden state h!



Attention weights p_i indicate features that the model gives the most importance to at every time-step

Soft attention in RNNs

Note that f_A produces attention weights as a function of both current input x as well as previous hidden state h!



Attention weights p_i indicate features that the model gives the most importance to at every time-step i

Weight matrix A shared across multiple timesteps in computational graph

Active areas of research

- Improving prediction models for clinically meaningful tasks

Active areas of research

- Improving prediction models for clinically meaningful tasks
  - Another popular task: early warning for critical conditions such as sepsis

Active areas of research

- Improving prediction models for clinically meaningful tasks
  - Another popular task: early warning for critical conditions such as sepsis
  - Multimodal modeling: more effective joint reasoning over different modalities of data (e.g. text, lab results, images, etc.)

Summary

Today's topics

- More on EHR data, missing values, and data formats
- More on feature representations
- A first look at model interpretability: soft attention

Next lecture

- More on text data and representations