Lecture 7:
Electronic Health Records
(Part 2)

Announcements

- Upcoming deadlines:
  - A1 due tomorrow, Oct 6
  - Project proposal due Fri, Oct 9
    - Remember that you must **train** a deep learning model somewhere in your project!
- A2 will be released Wed Oct 7, due Wed Oct 21 **(note change to Wed schedule)**
- Please consider posting homework questions visible to the entire class when appropriate -- everyone will benefit
- Please be careful of your GCP credits usage -- use the cheapest GPU feasible for each part of the assignment (this is specified in the assignment), and turn off your instance when not working on your hw

Last time: overview of electronic health records

Patient chart in digital form, containing medical and treatment history

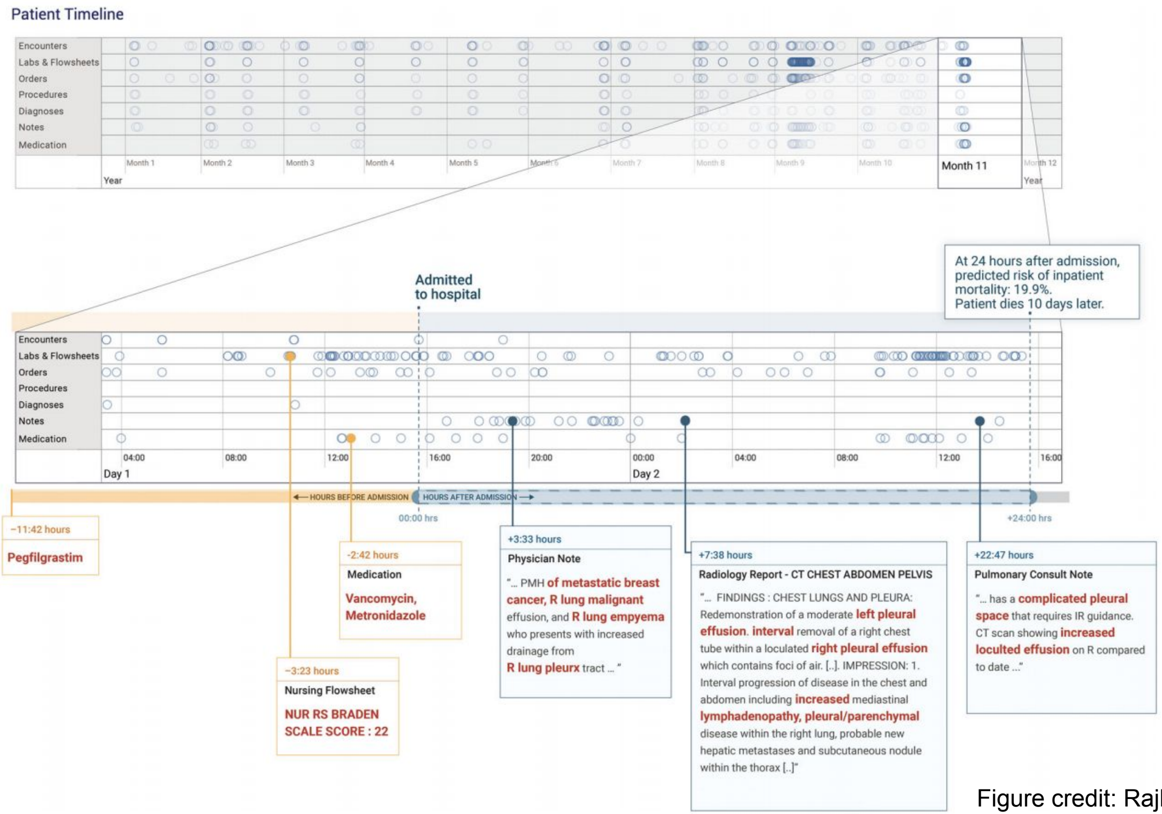
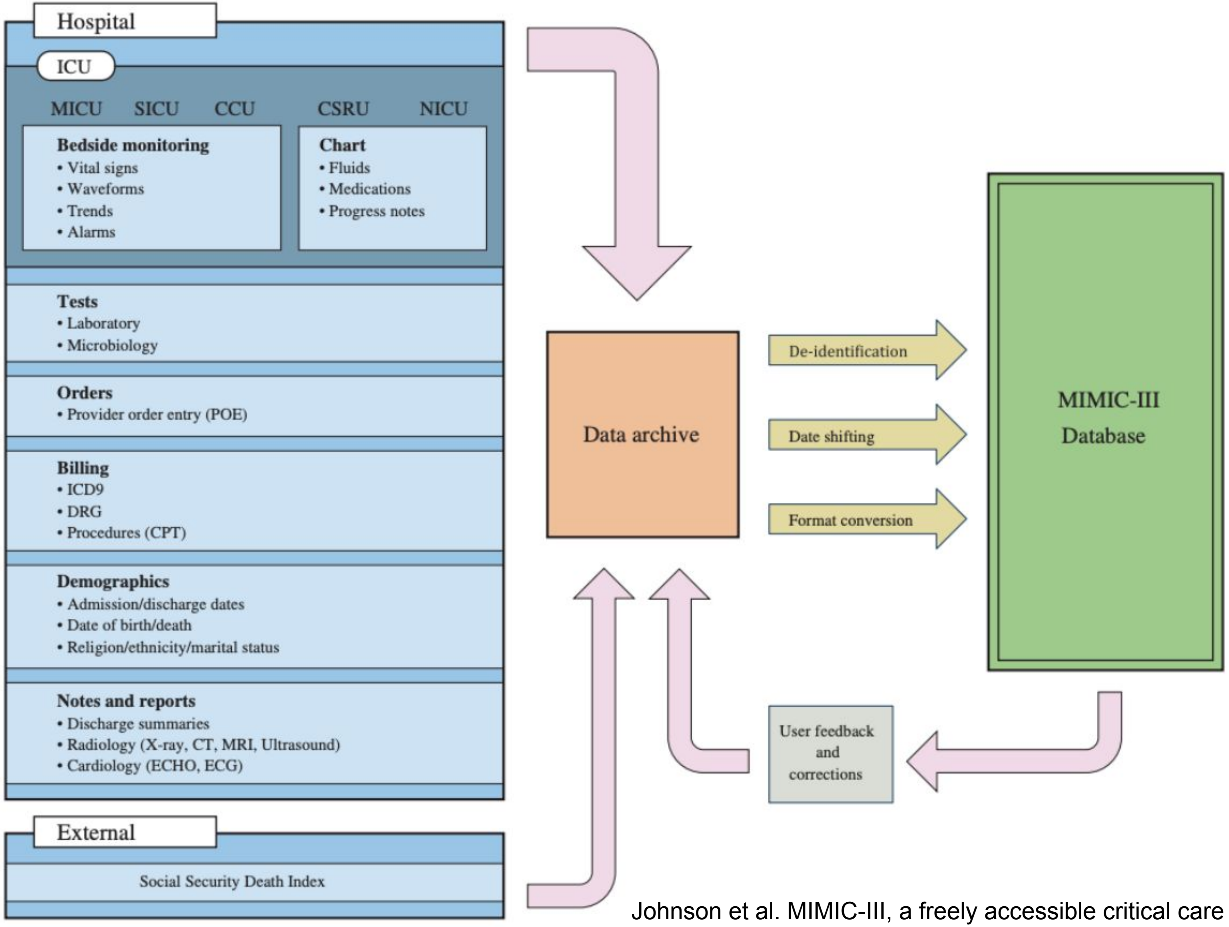


Figure credit: Rajkomar et al. 2018

A real example of EHR data: MIMIC-III dataset



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

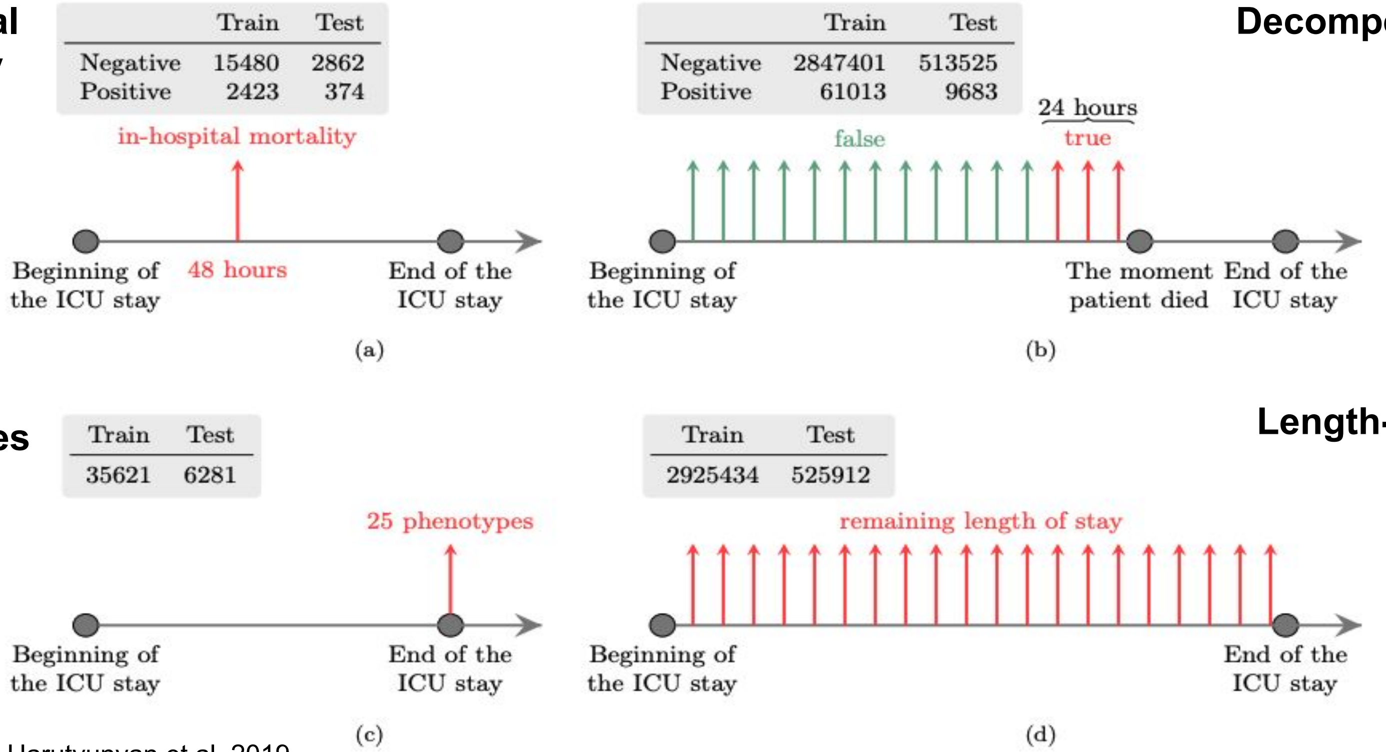
Examples of prediction tasks

In-hospital mortality

Decompensation

Phenotypes

Length-of-stay



In-hospital mortality:

	Train	Test
Negative	15480	2862
Positive	2423	374

Decompensation:

	Train	Test
Negative	2847401	513525
Positive	61013	9683

Phenotypes:

Train	Test
35621	6281

Length-of-stay:

Train	Test
2925434	525912

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:

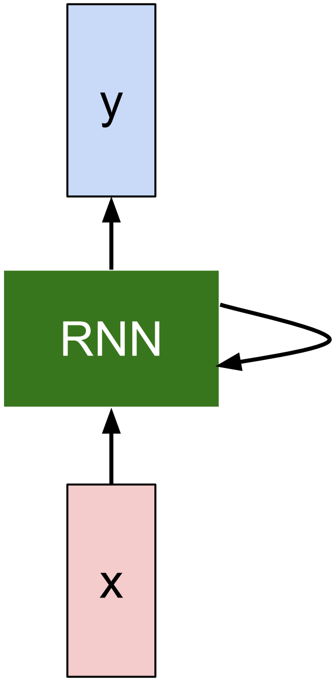
$$\boxed{h_t} = \boxed{f_W}(\boxed{h_{t-1}}, \boxed{x_t})$$

new state

old state

input vector at some time step

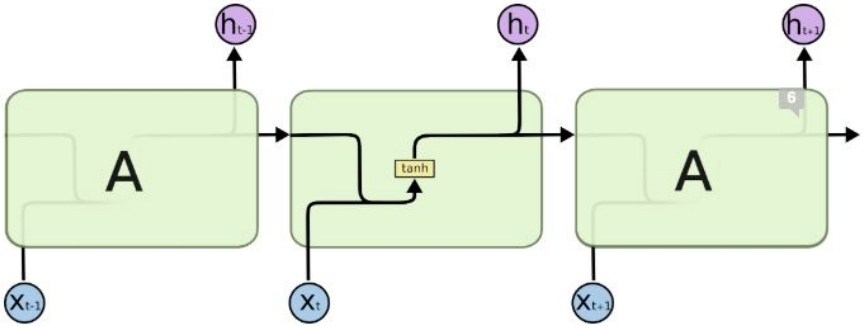
some function with parameters W



Slide credit: CS231n

Long Short Term Memory (LSTM) Recurrent Networks

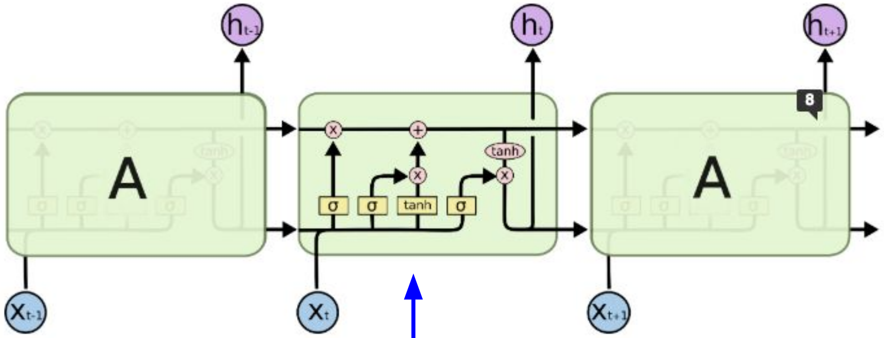
Unrolled Vanilla RNN



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

$$y_t = W_{hy}h_t$$

Unrolled LSTM



Different computation to obtain h_t

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

Today

- More on EHR data
- More on feature representations
- A first look at model interpretability: soft attention

Sources of EHR data

- Open-source EHR datasets (MIMIC-III, 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, 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

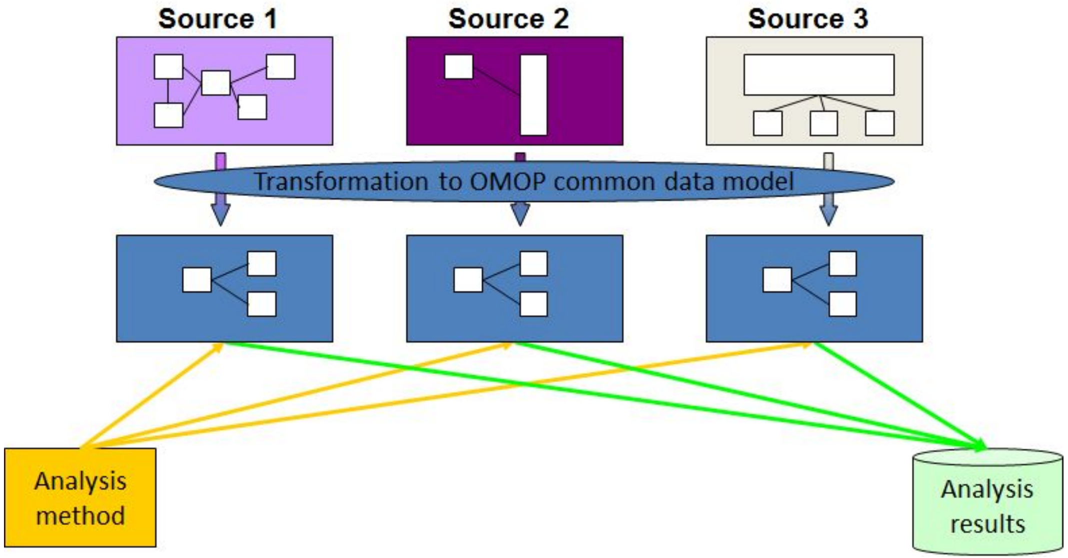


Figure credit: https://www.ohdsi.org/wp-content/uploads/2014/07/Why-CDM.png

OMOP Common Data Model

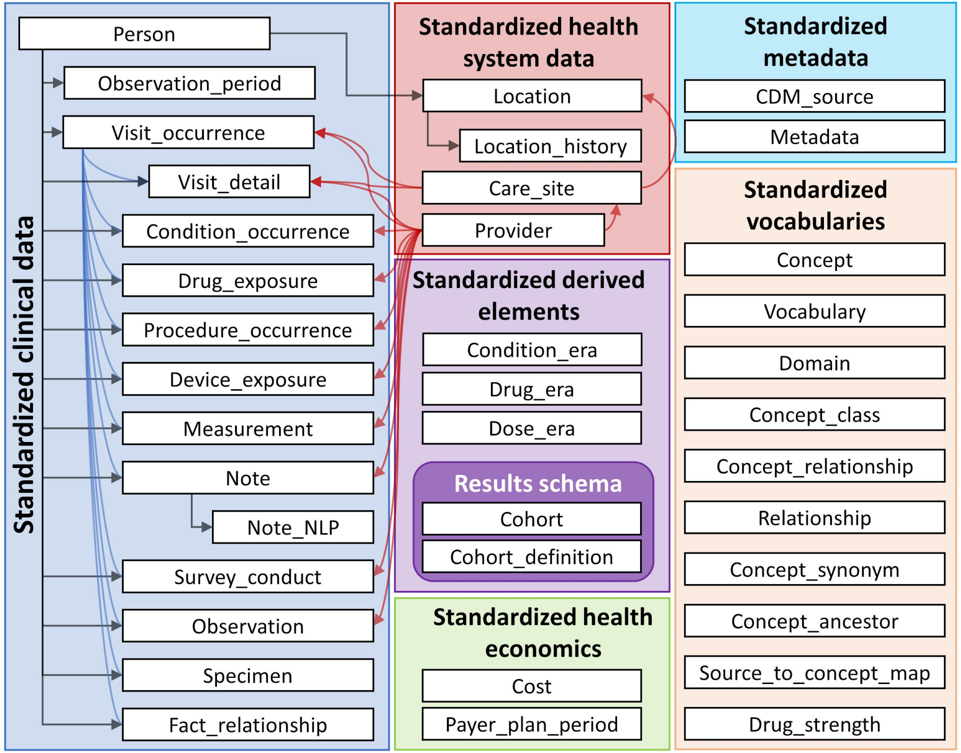


Figure credit: https://ohdsi.github.io/TheBookOfOhdsi/images/CommonDataModel/cdmDiagram.png

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

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

FHIR

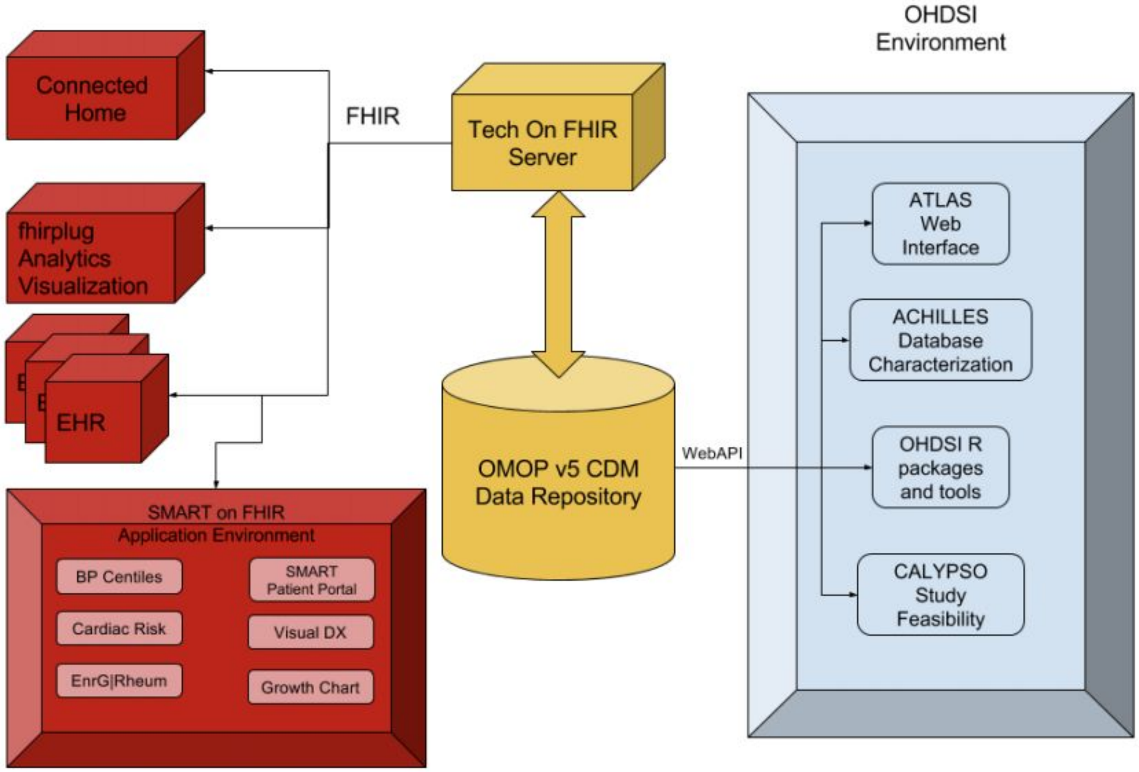


Figure credit: Choi et al. OHDSI on FHIR Platform Development with OMOP CDM mapping to FHIR Resources. 2016.

FHIR

FHIR-based information exchange between different sources

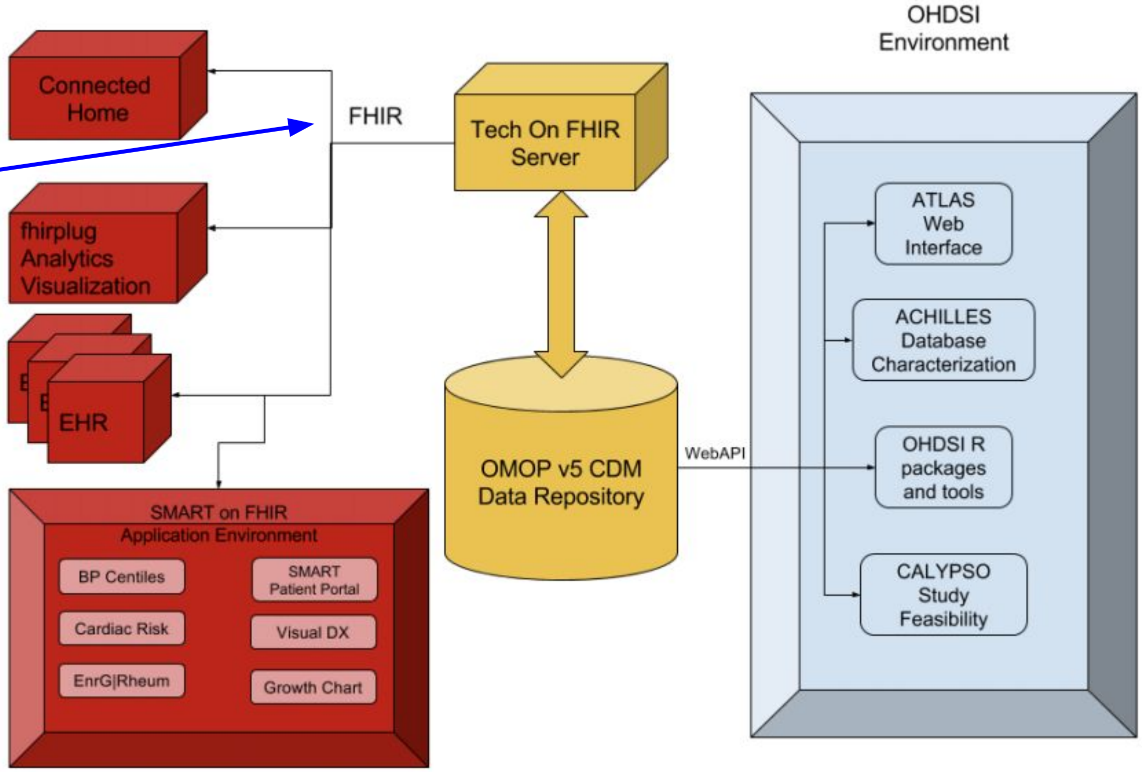


Figure credit: Choi et al. OHDSI on FHIR Platform Development with OMOP CDM mapping to FHIR Resources. 2016.

FHIR

Data from all sources can be written in an OMOP data repository for analysis

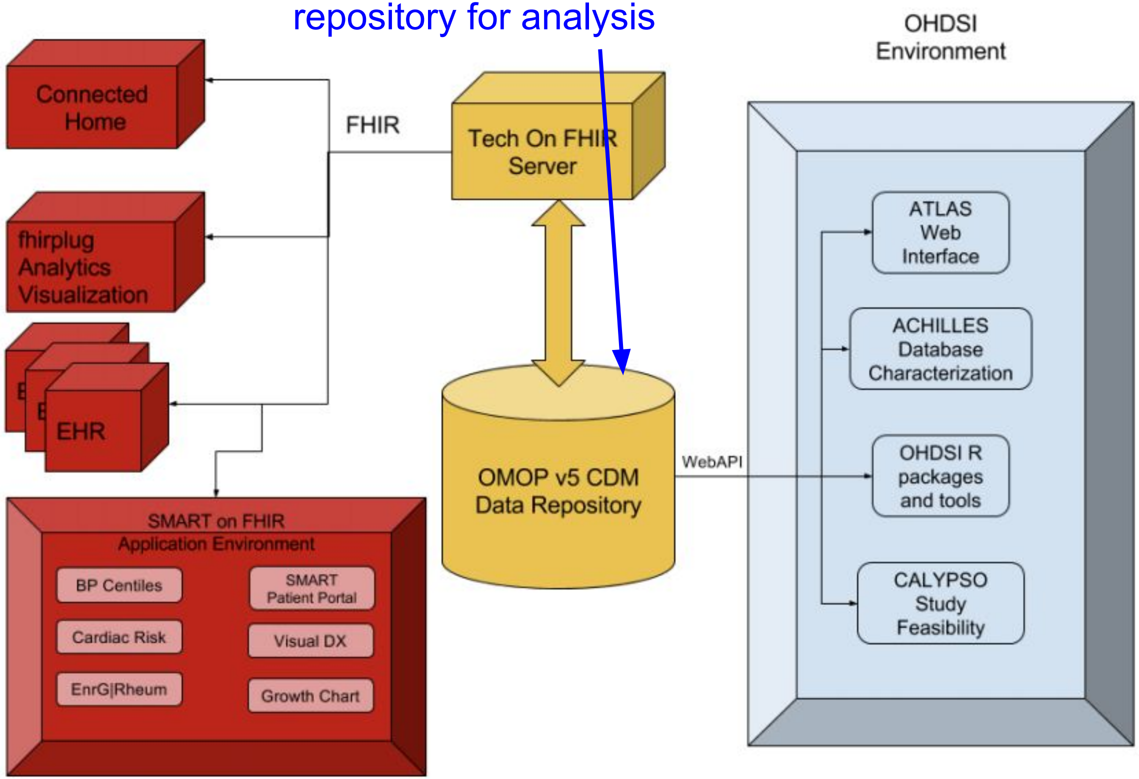


Figure credit: Choi et al. OHDSI on FHIR Platform Development with OMOP CDM mapping to FHIR Resources. 2016.

FHIR

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

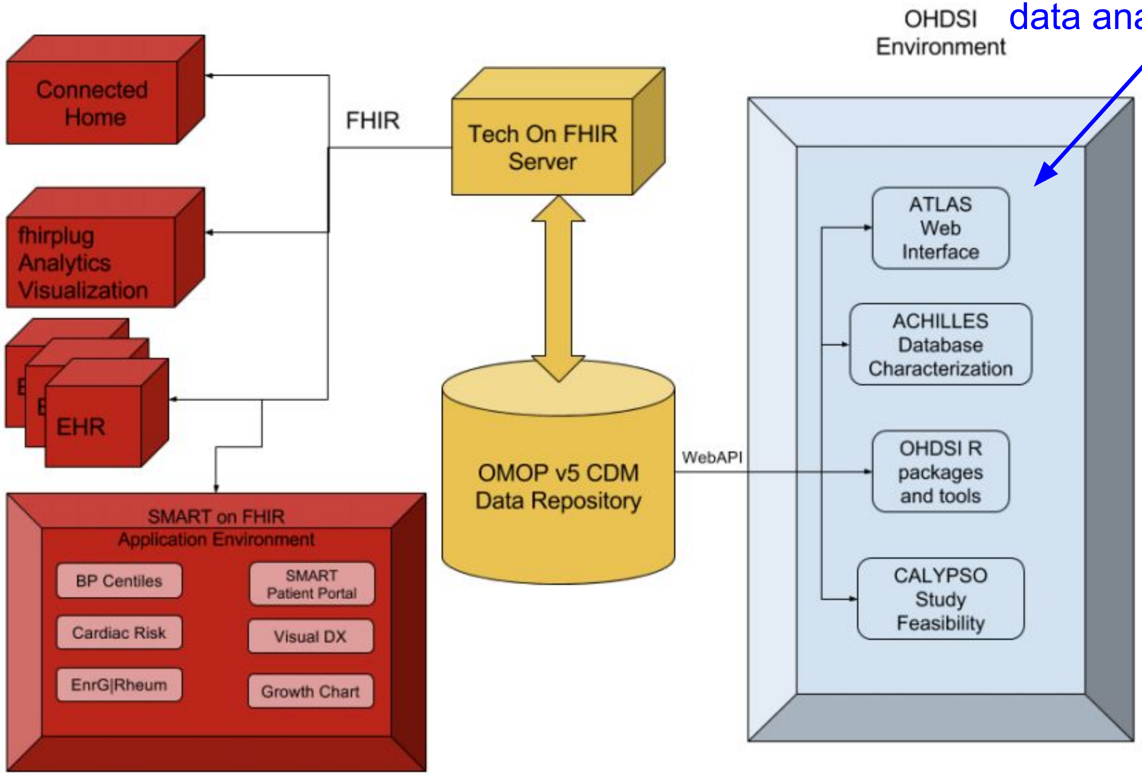
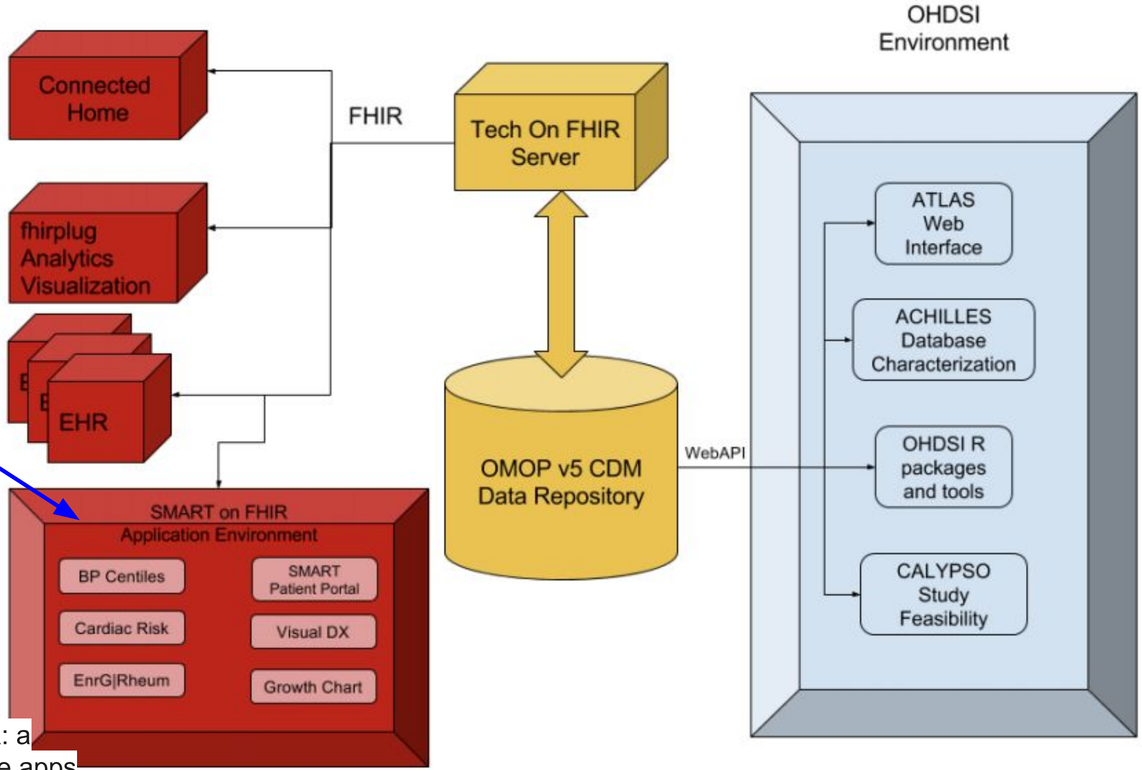


Figure credit: Choi et al. OHDSI on FHIR Platform Development with OMOP CDM mapping to FHIR Resources. 2016.

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.

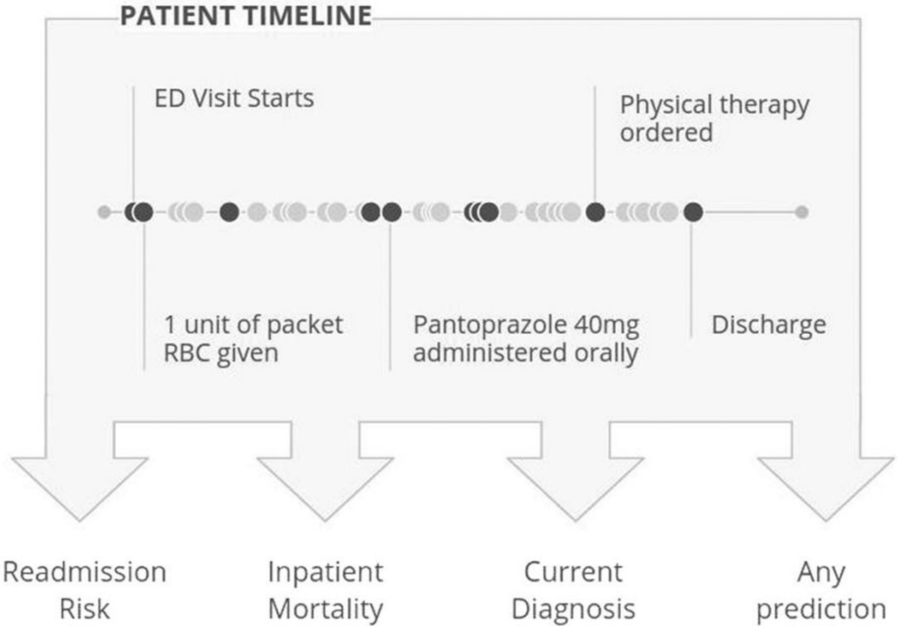
Figure credit: Choi et al. OHDSI on FHIR Platform Development with OMOP CDM mapping to FHIR Resources. 2016.

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

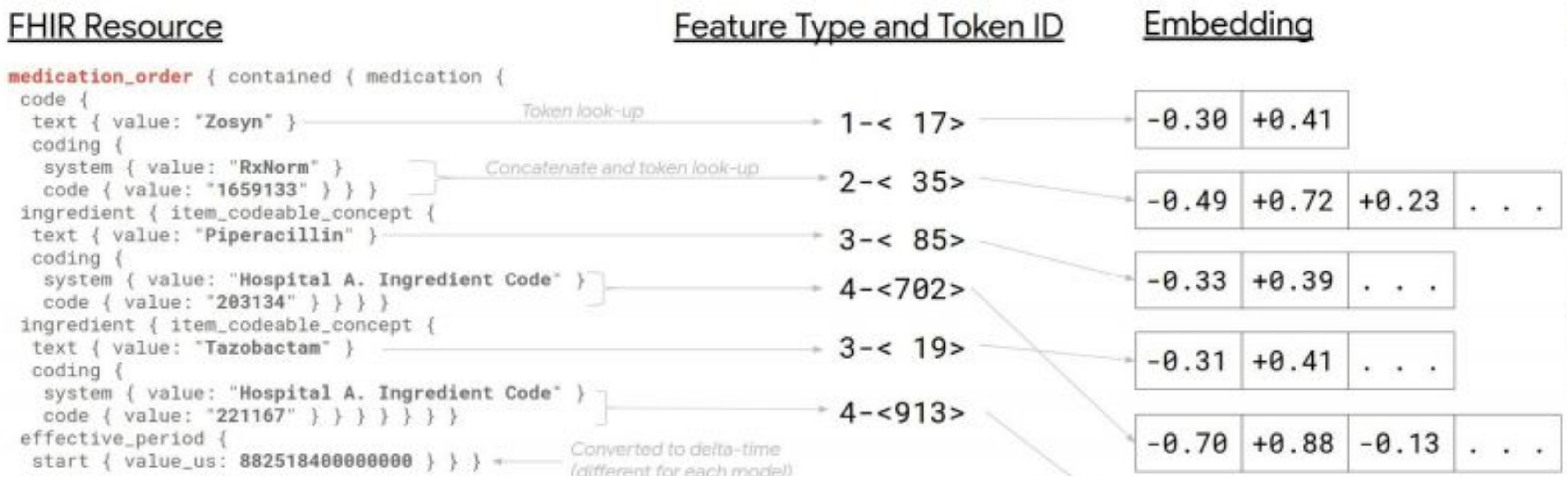
Rajkomar et al. 2018

- 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

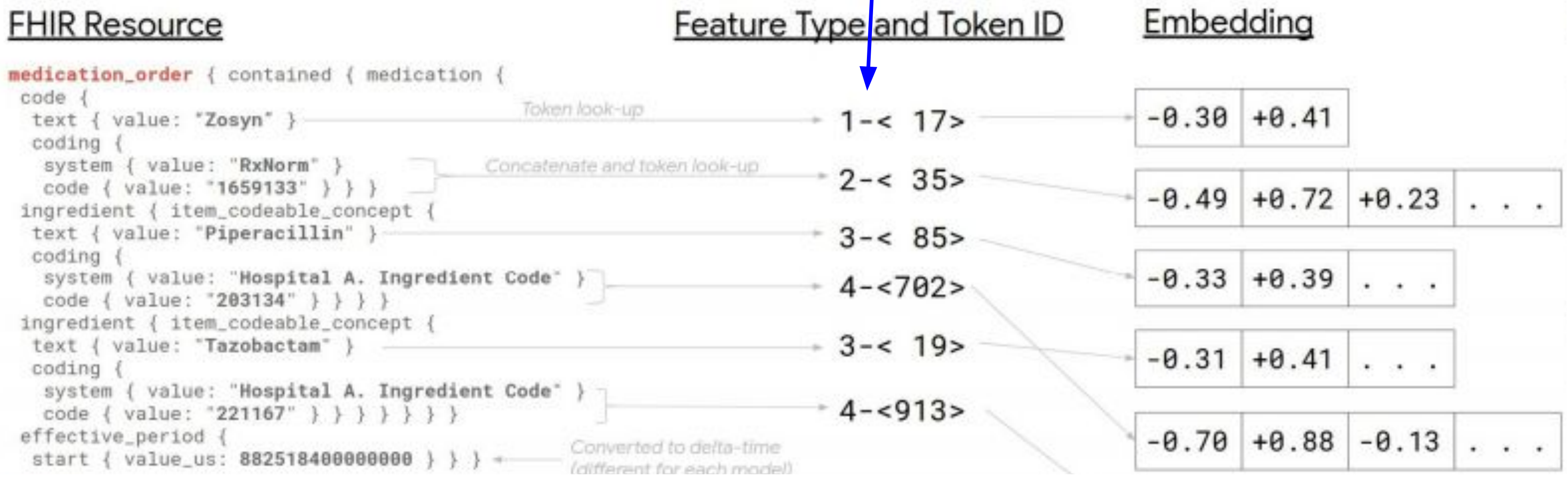


Raw data as FHIR resources

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

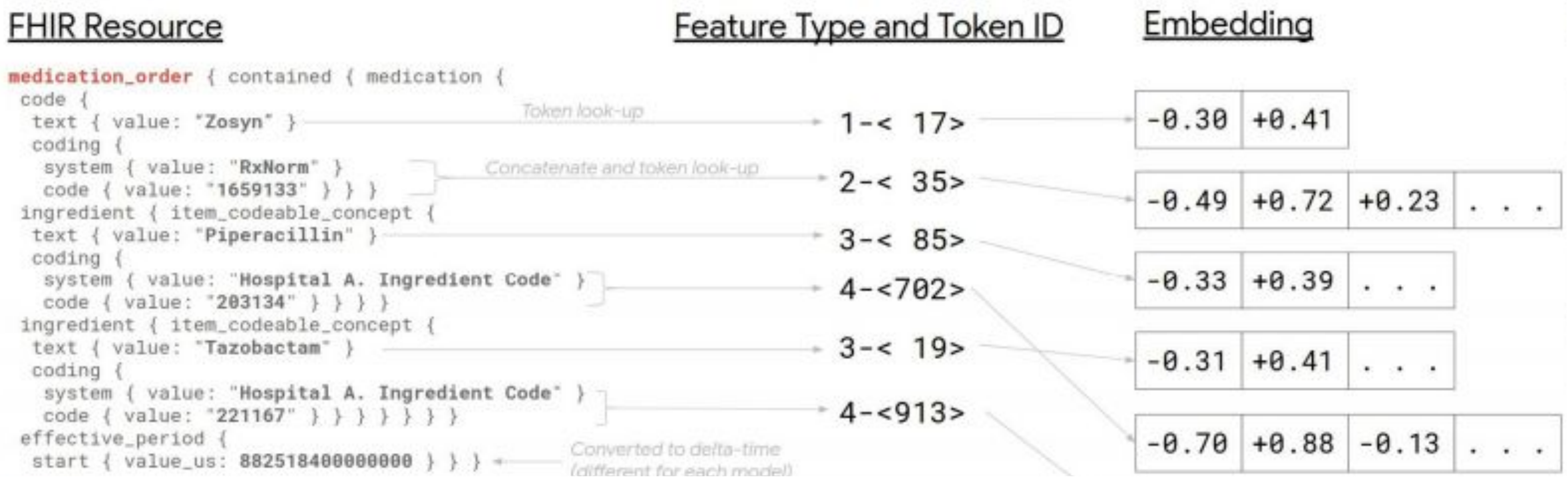
Data representation

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

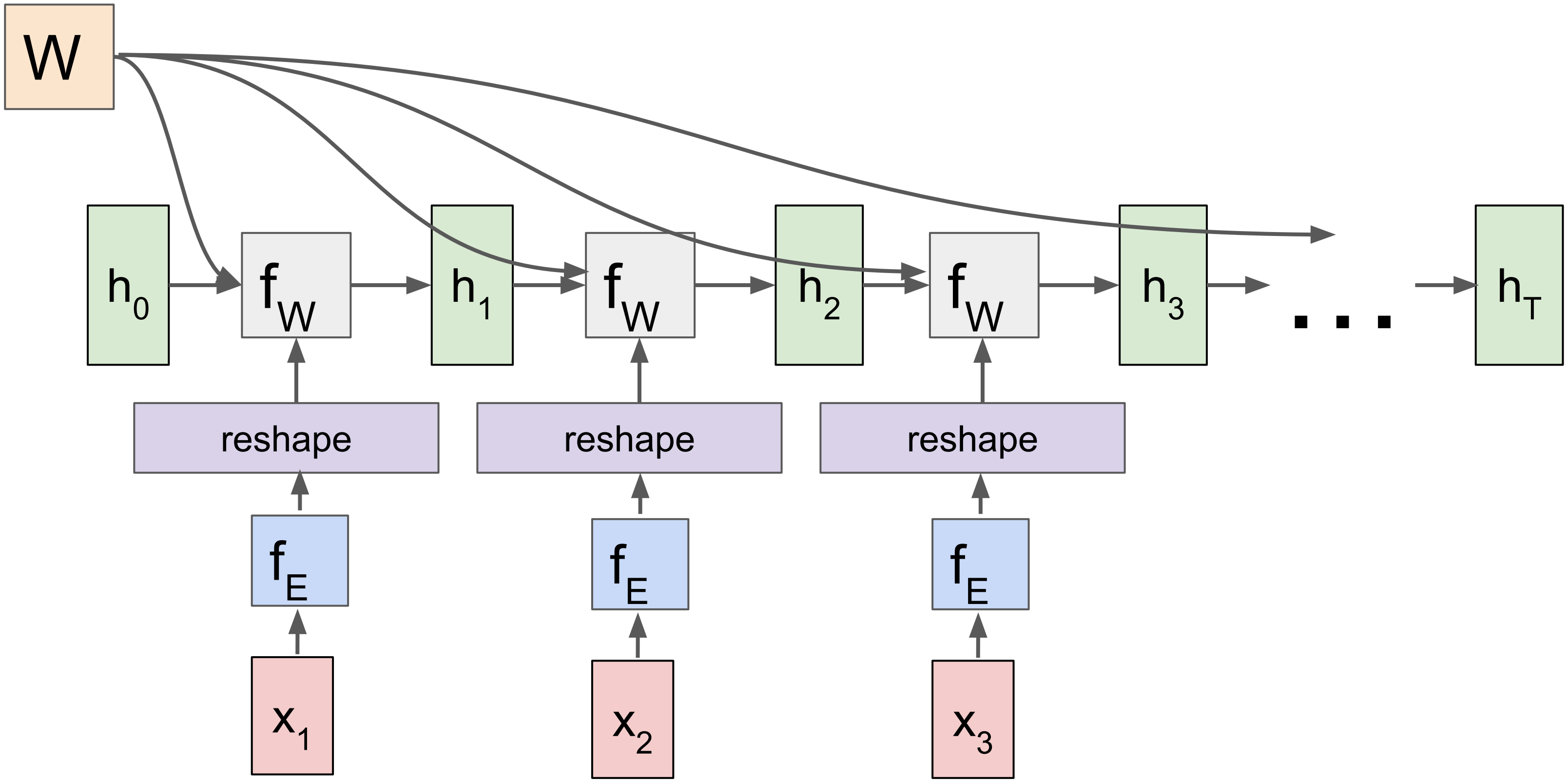
N x D embedding matrix

= [0.5   0.8   0.2]

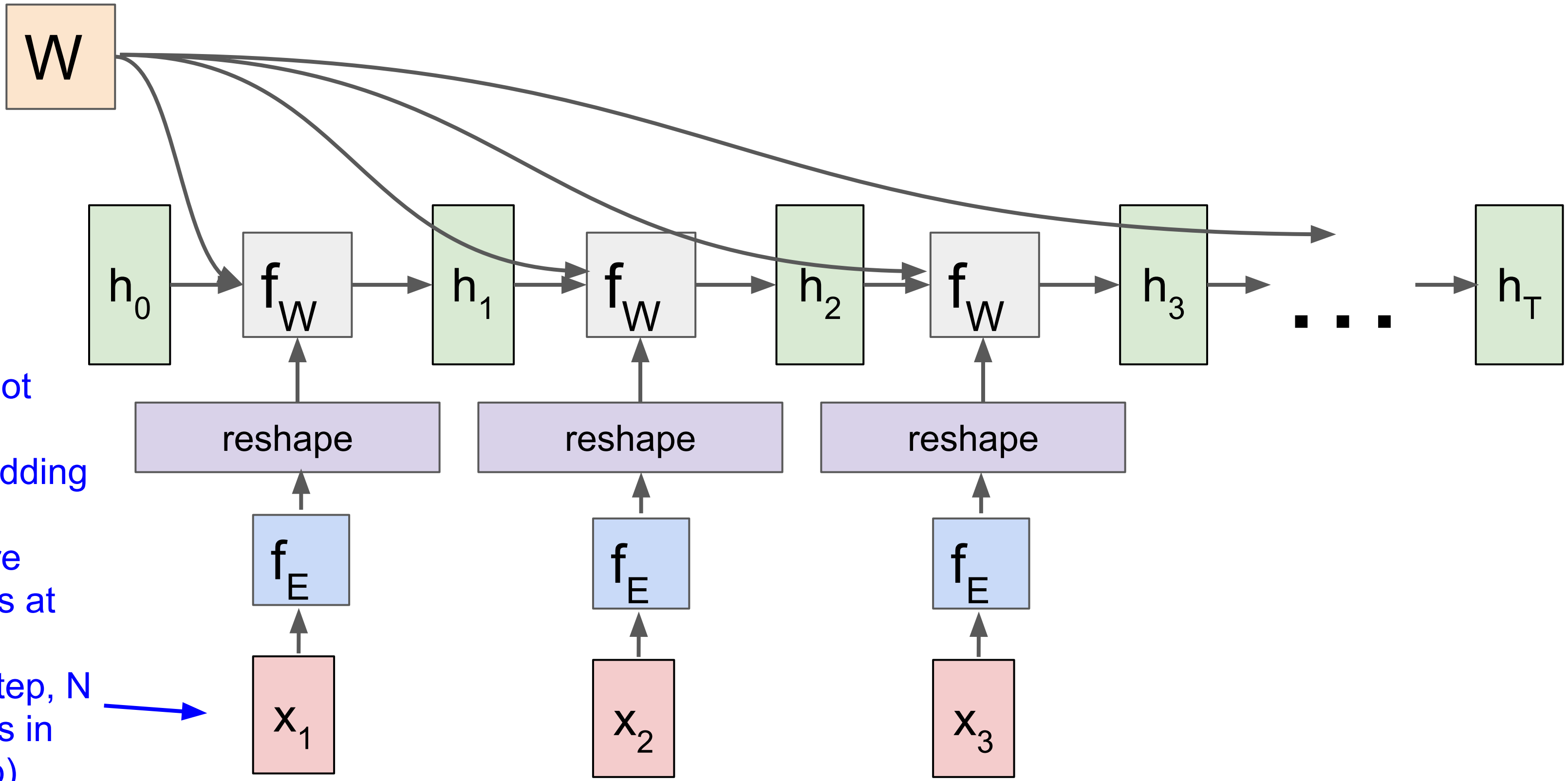
D-dim token embedding

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



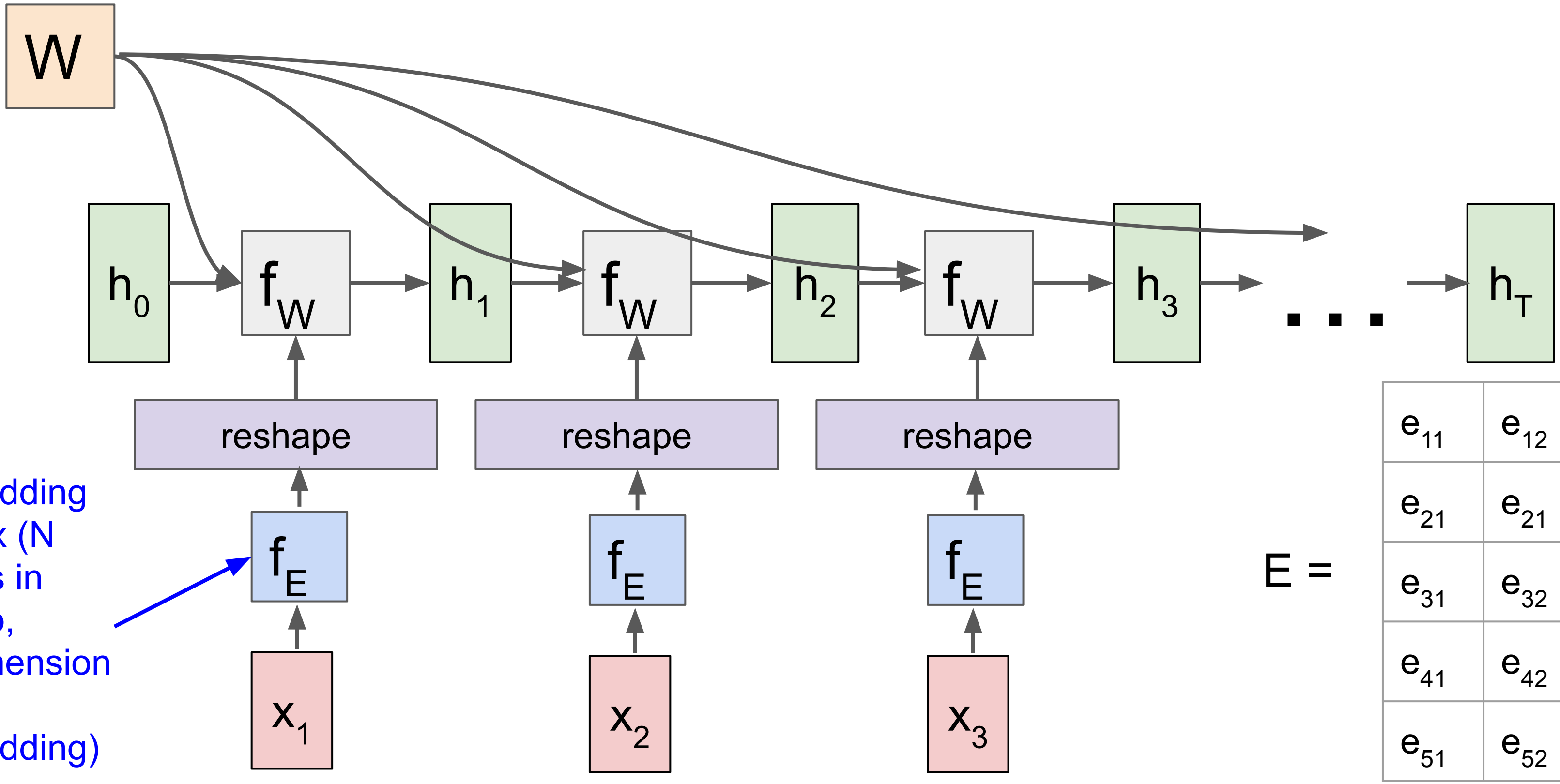
Computational graph input to RNN



M x N one-hot token embeddings (M feature tokens at the timestep, N tokens in vocab)

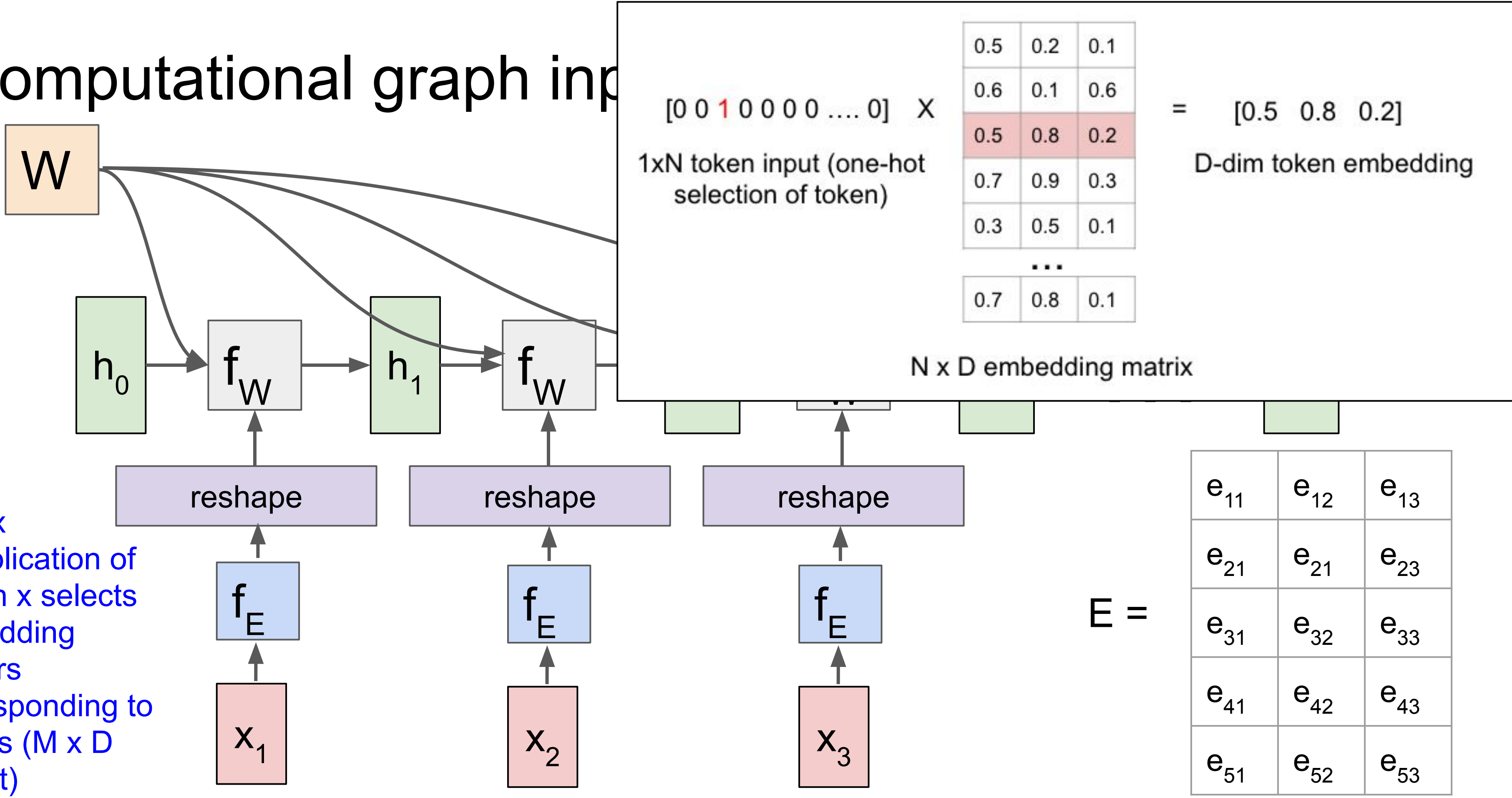
Computational graph input to RNN

N x D embedding matrix (N words in vocab, D-dimensional embedding)



$$E = \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{21} & e_{23} \\ e_{31} & e_{32} & e_{33} \\ e_{41} & e_{42} & e_{43} \\ e_{51} & e_{52} & e_{53} \end{bmatrix}$$

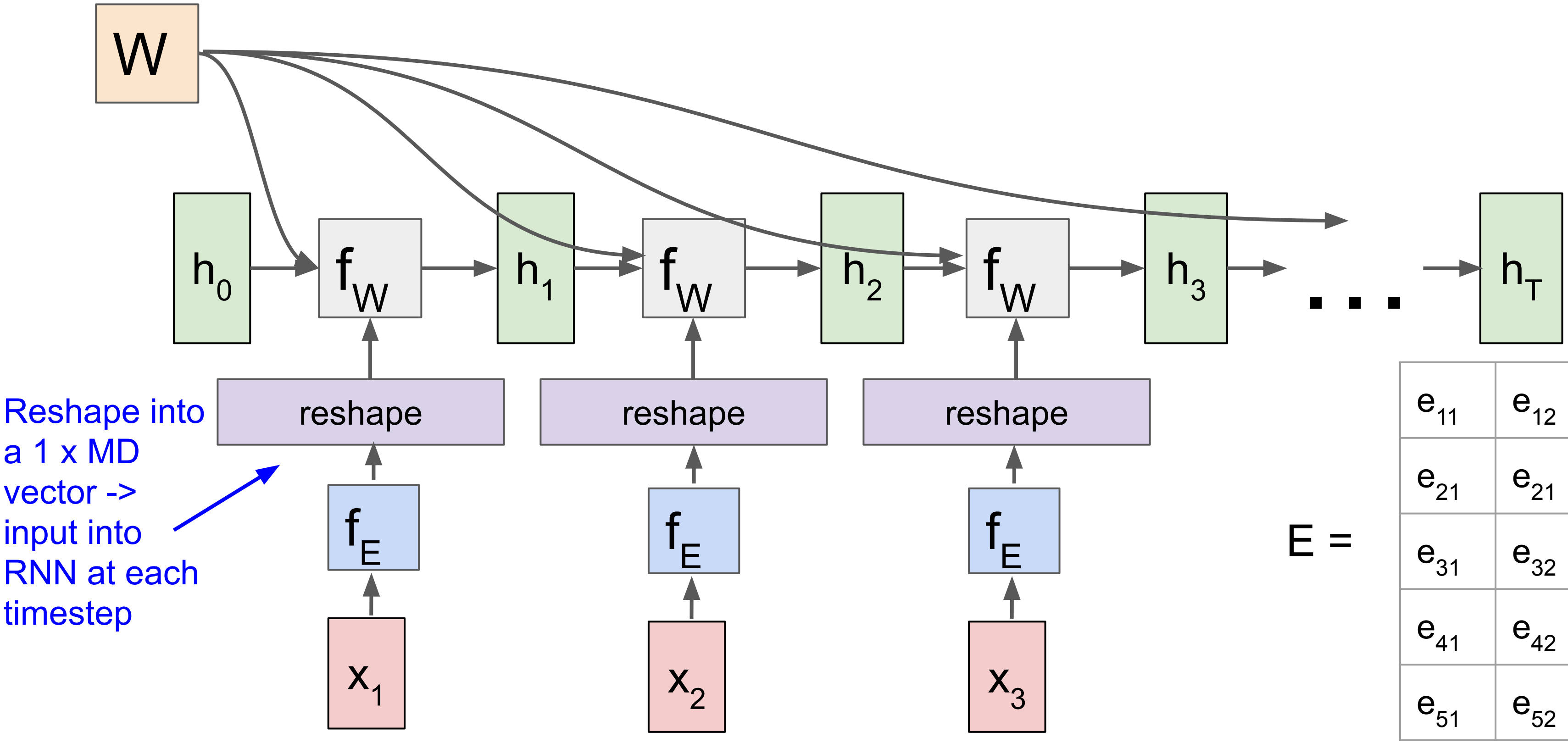
Computational graph inp



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

$$E = \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{21} & e_{23} \\ e_{31} & e_{32} & e_{33} \\ e_{41} & e_{42} & e_{43} \\ e_{51} & e_{52} & e_{53} \end{bmatrix}$$

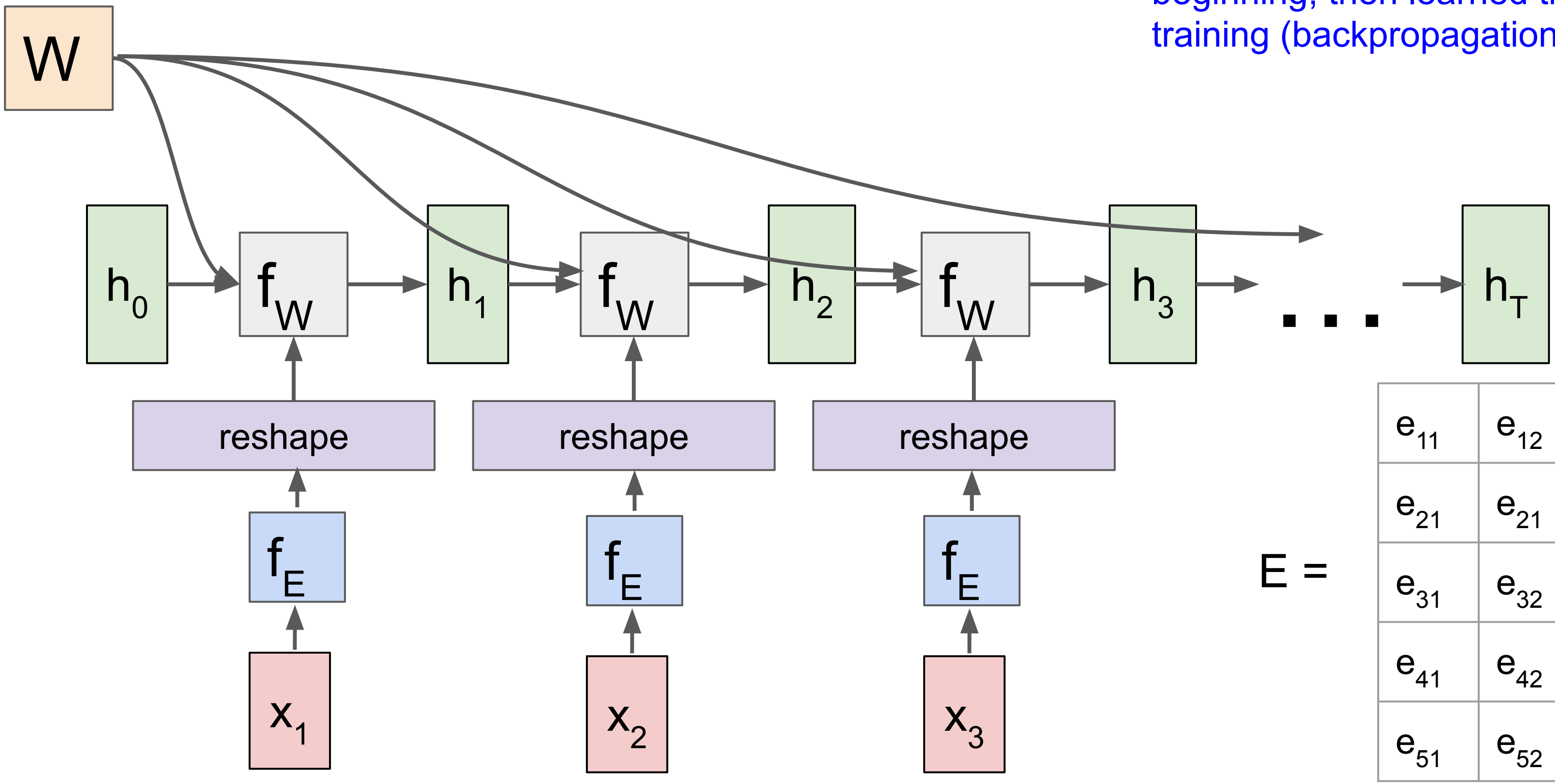
Computational graph input to RNN



$$E = \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{21} & e_{23} \\ e_{31} & e_{32} & e_{33} \\ e_{41} & e_{42} & e_{43} \\ e_{51} & e_{52} & e_{53} \end{bmatrix}$$

Computational graph input to RNN

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

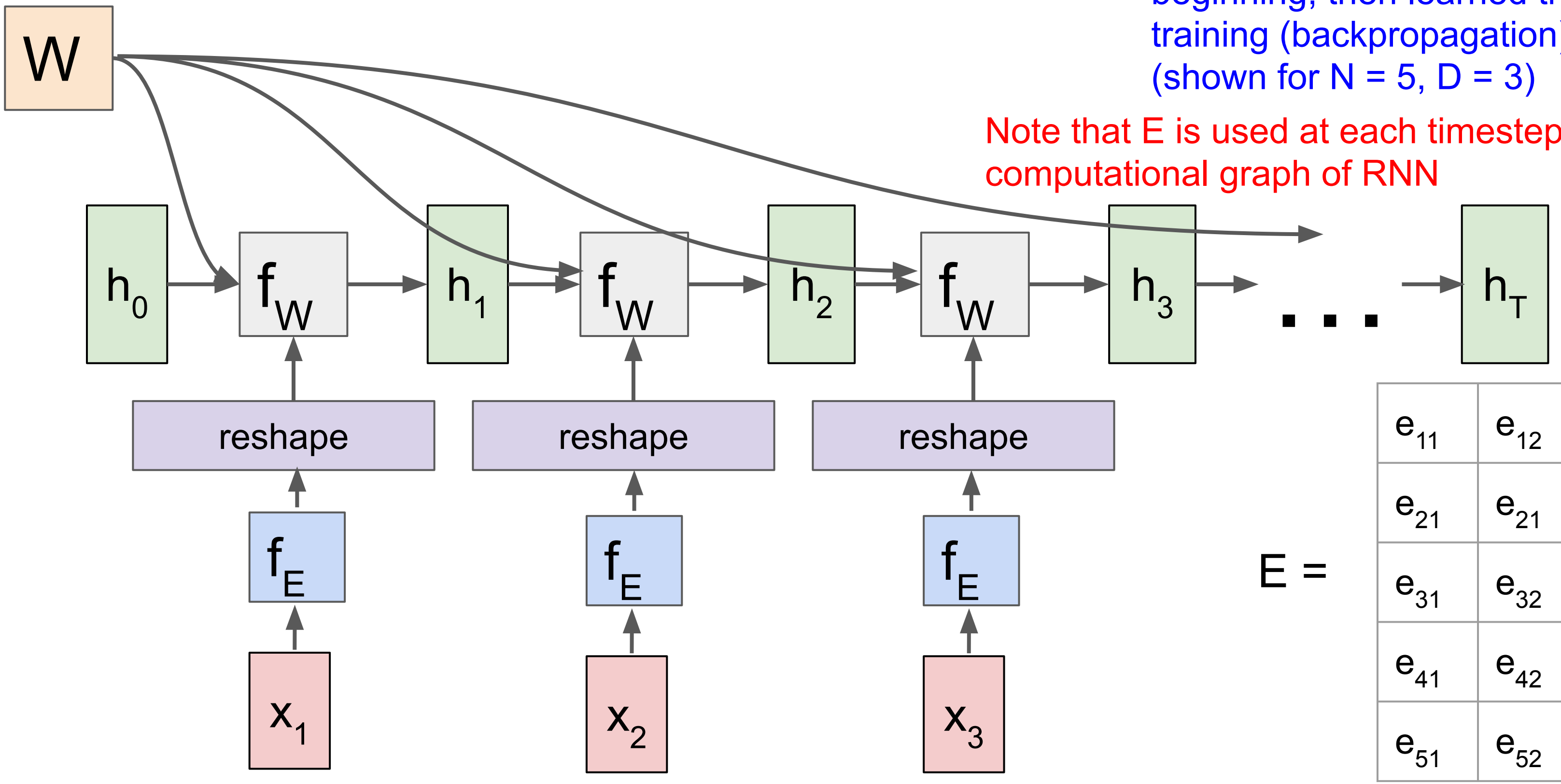


$$E = \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{21} & e_{23} \\ e_{31} & e_{32} & e_{33} \\ e_{41} & e_{42} & e_{43} \\ e_{51} & e_{52} & e_{53} \end{bmatrix}$$

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

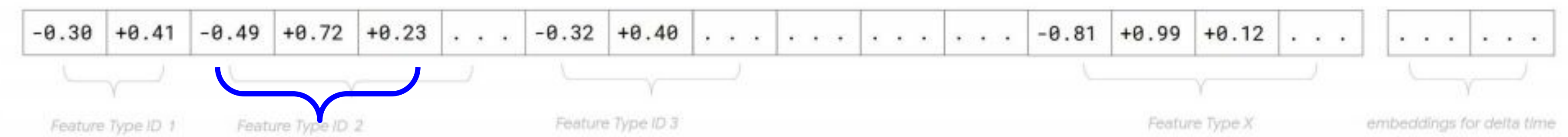


$$E = \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{21} & e_{23} \\ e_{31} & e_{32} & e_{33} \\ e_{41} & e_{42} & e_{43} \\ e_{51} & e_{52} & e_{53} \end{bmatrix}$$

Rajkomar et al. RNN (LSTM) input

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

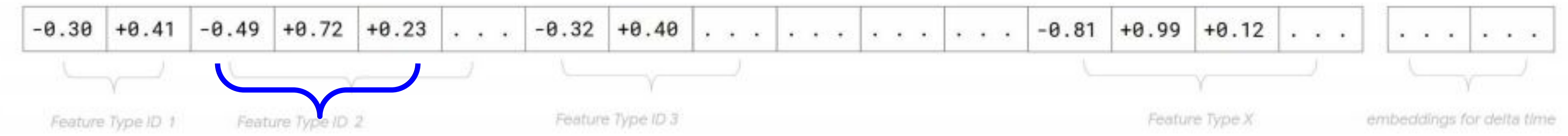
Rajkomar et al. RNN (LSTM) input



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



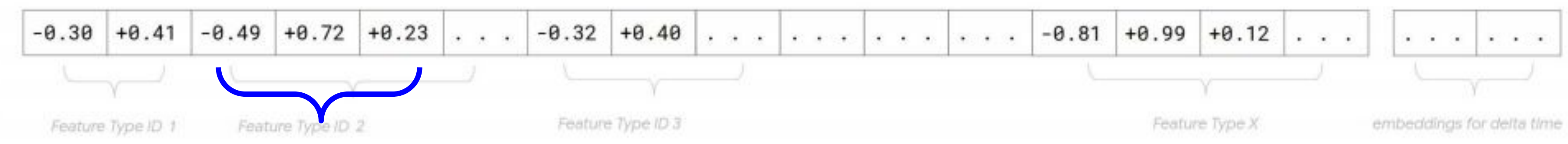
One vector representation for each token "feature type" (e.g. medication, procedure). Embeddings of multiple tokens corresponding to a same feature type are combined through averaging.

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. Full high-temporal resolution data are bucketed and (weight-)averaged into coarser buckets for RNN input.



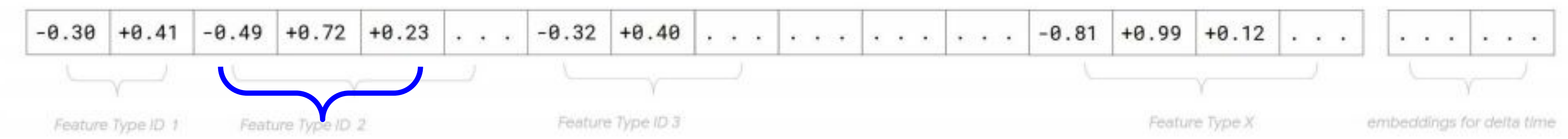
One vector representation for each token "feature type" (e.g. medication, procedure). Embeddings of multiple tokens corresponding to a same feature type are combined through averaging.

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. Full high-temporal resolution data are bucketed and (weight-)averaged into coarser buckets for 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

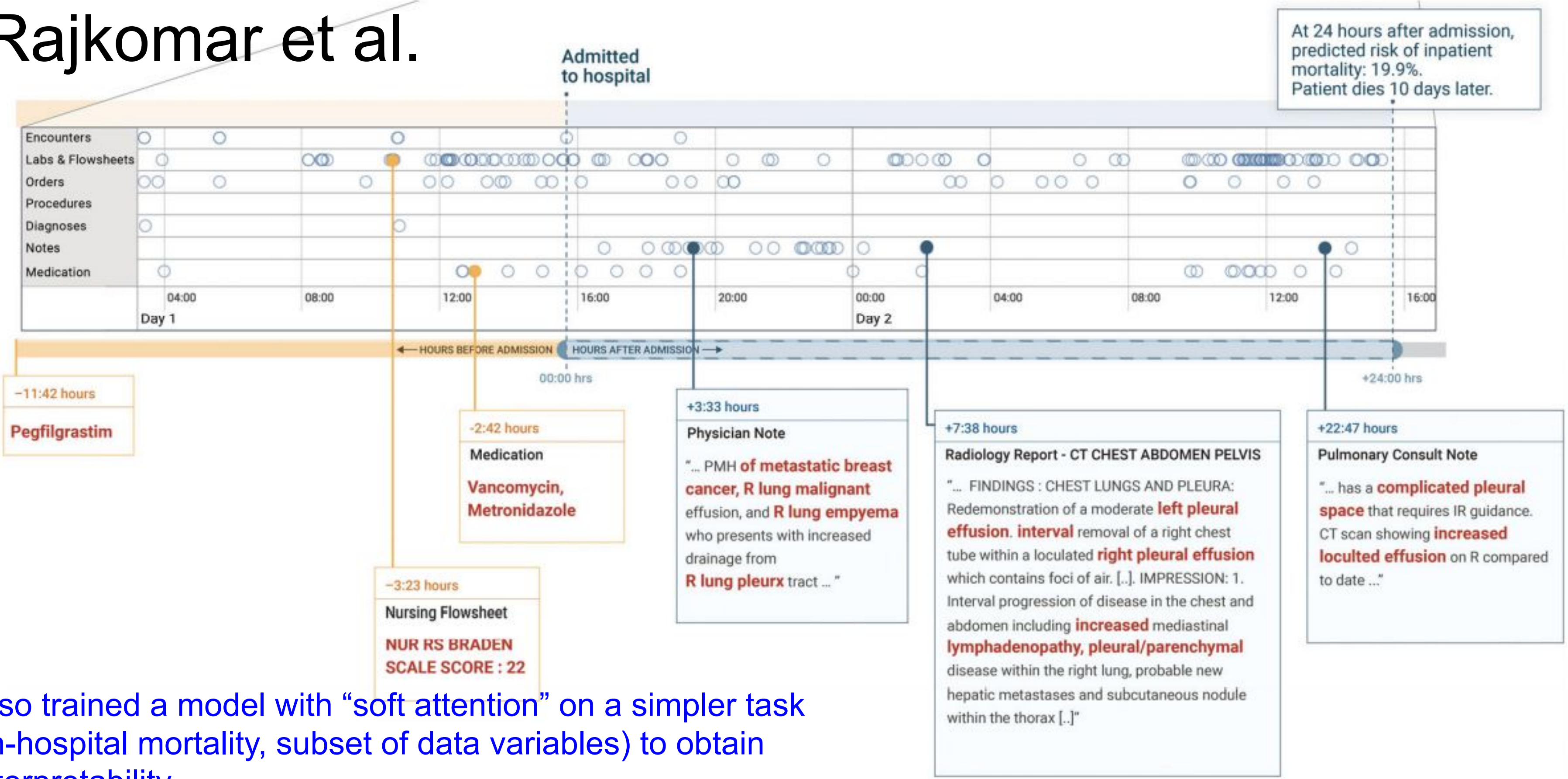
Evaluated model at 24 hr before admission, at admission, and 24 hr after admission

	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.

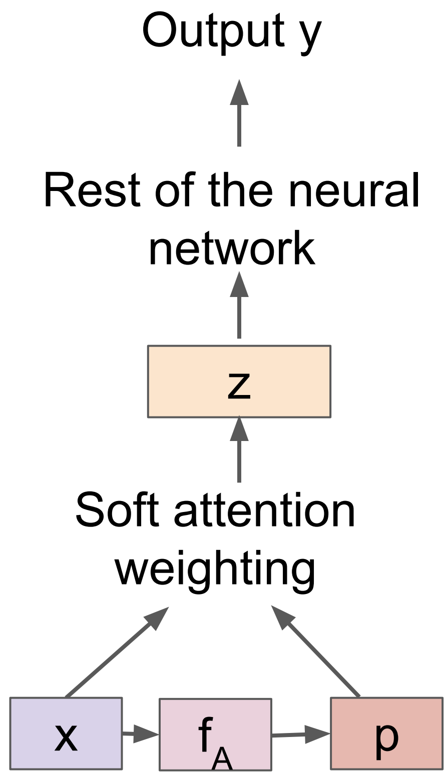


Also trained a model with "soft attention" on a simpler task (in-hospital mortality, subset of data variables) to obtain interpretability

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

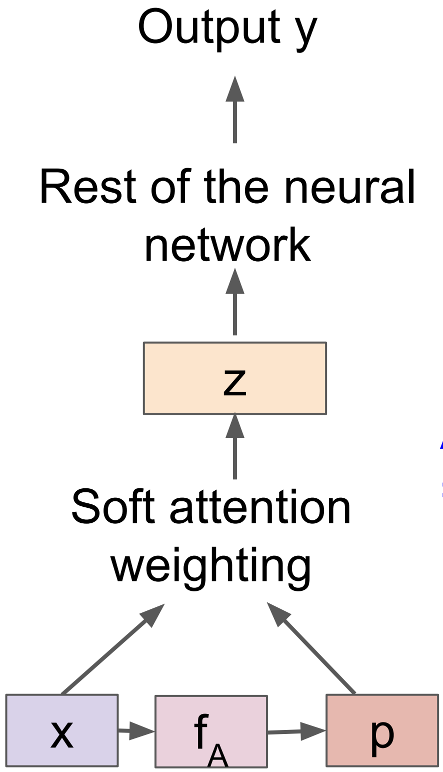
Soft attention

- 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 $p = [p_1, p_2, \ldots, p_D]$

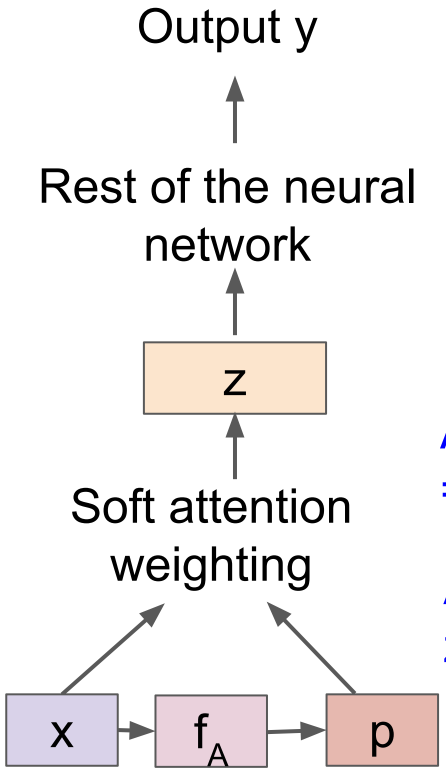
Attention-weighted input $z = [z_1, z_2, \ldots, z_D]$

Learnable fully connected layer f_A with weights A

Soft attention

$$p = \mathrm{softmax}(Ax)$$

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

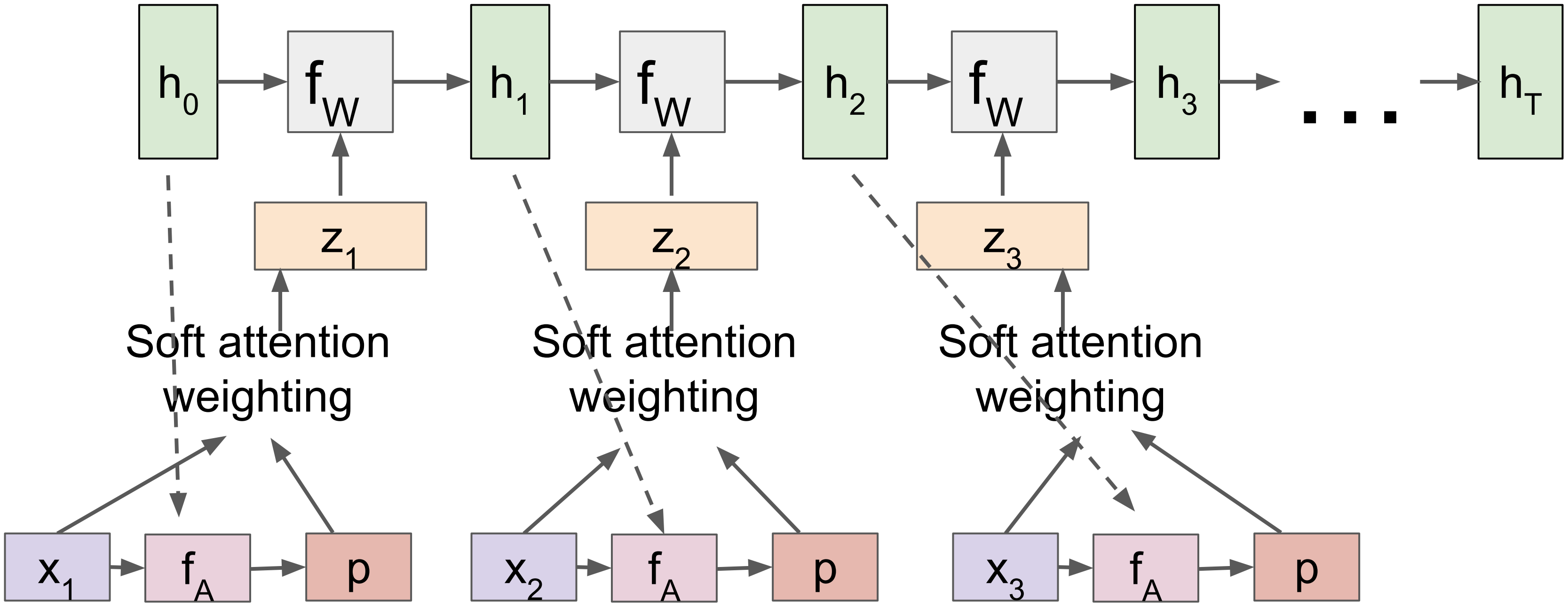
Attention weights $p = [p_1, p_2, \ldots, p_D]$

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

Learnable fully connected layer f_A with weights A

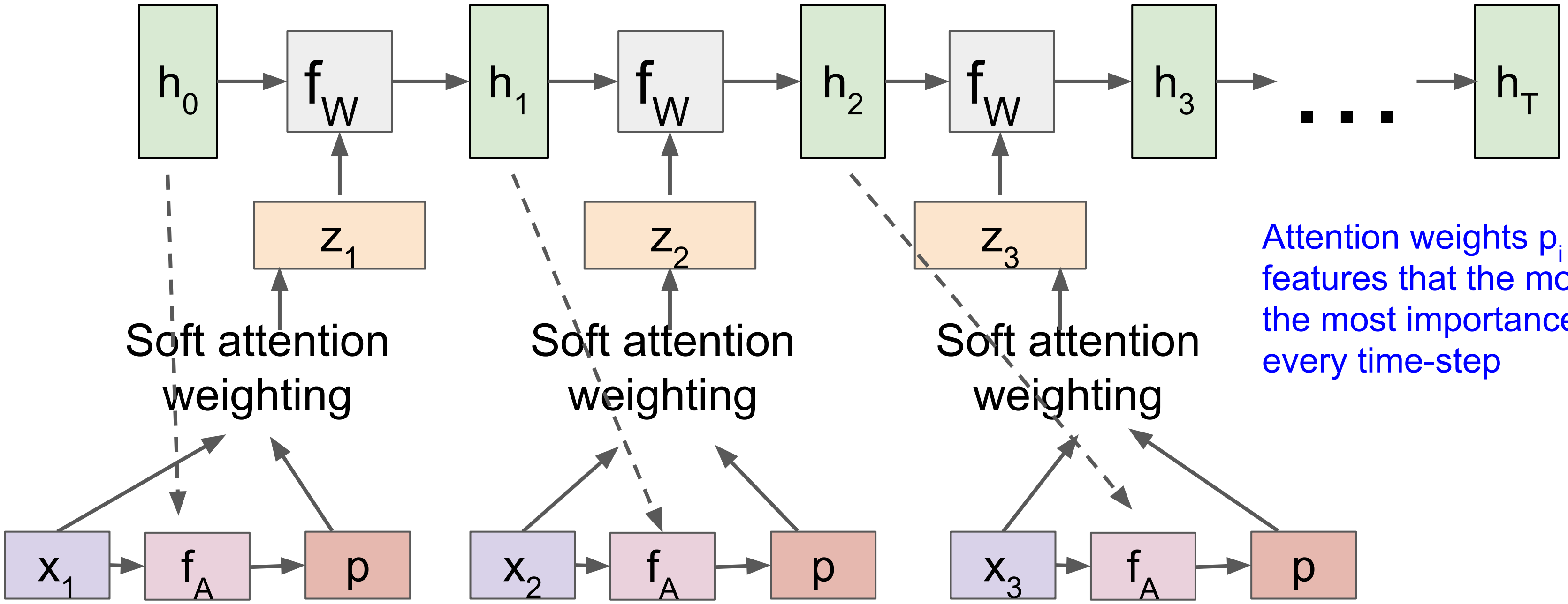
Soft attention in RNNs

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

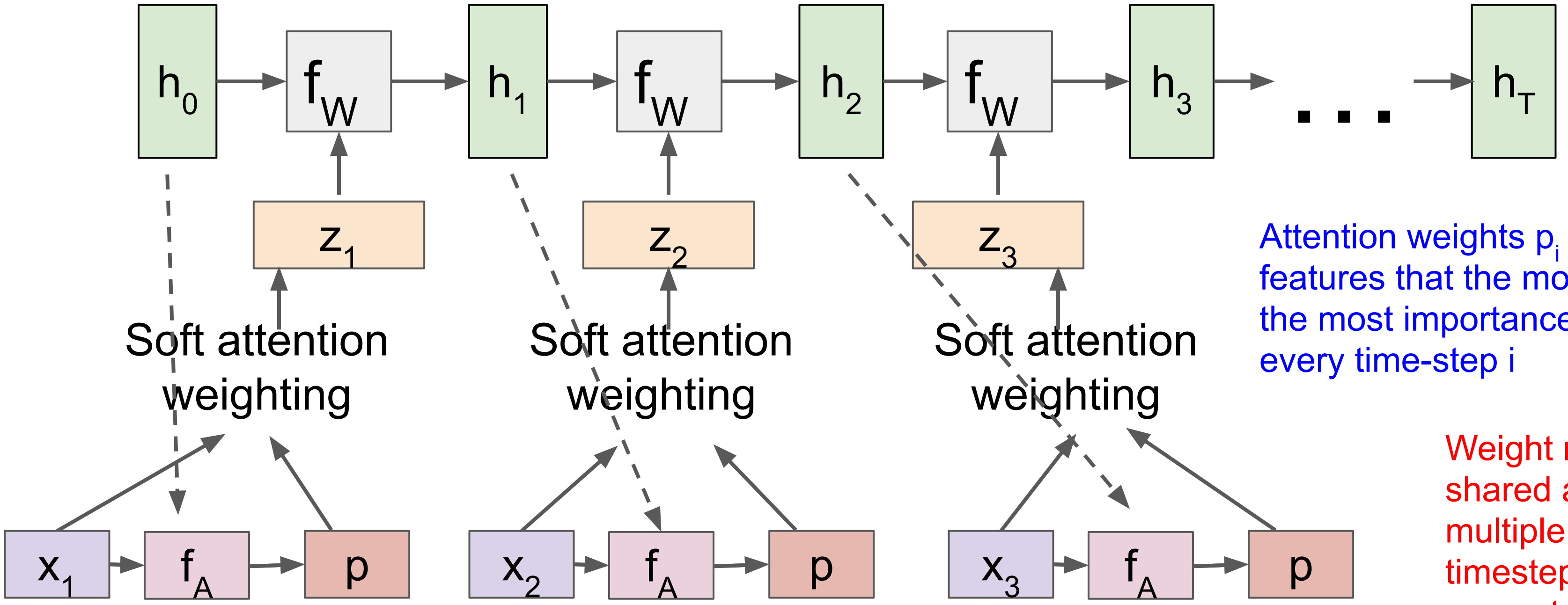
Note that f_A produces attention weights as a function of both current input x as well as previous hidden state h!



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.)

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.)
- Model interpretability

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.)
- Model interpretability
- Learning useful feature representations for downstream tasks

-0.30	+0.41	-0.49	+0.72	+0.23	. . .	-0.32	+0.40	. . .	. . .	. . .	. . .	-0.81	+0.99	+0.12	. . .	. . .	. . .

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.)
- Model interpretability
- Learning useful feature representations for downstream tasks
    - E.g., unsupervised word embedding methods from NLP for clinical notes -> next lecture

-0.30	+0.41	-0.49	+0.72	+0.23	. . .	-0.32	+0.40	. . .	. . .	. . .	. . .	-0.81	+0.99	+0.12	. . .	. . .	. . .

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.)
- Model interpretability
- Learning useful feature representations for downstream tasks
  - E.g., unsupervised word embedding methods from NLP for clinical notes -> next lecture
  - Another task, beyond prediction: finding cohorts of similar patients ("precision medicine")

-0.30	+0.41	-0.49	+0.72	+0.23	. . .	-0.32	+0.40	. . .	. . .	. . .	. . .	-0.81	+0.99	+0.12	. . .	. . .	. . .

Summary

Today's topics

- More on EHR data
- More on feature representations
- A first look at model interpretability: soft attention

Next lecture

- More on text data and representations