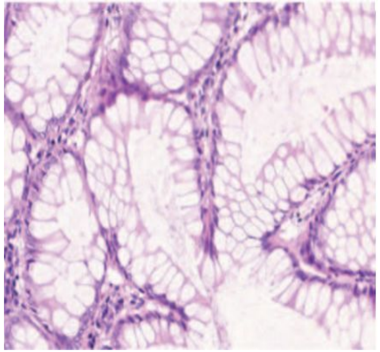
Lecture 5:
Electronic Health Records
Introduction

Announcements

- A1 due Tue 10/18
- Project proposal due Fri 10/21
- Project partner finding session during review section this Friday, 1:30pm, Alway M106

Last Time:
Richer visual recognition tasks: segmentation and detection

Classification

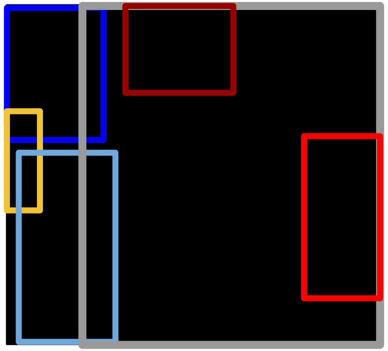


Output:
one category label for image (e.g., colorectal glands)

Semantic Segmentation

Output:
category label for each pixel in the image

Detection



Output:
Spatial bounding box for each **instance** of a category object in the image

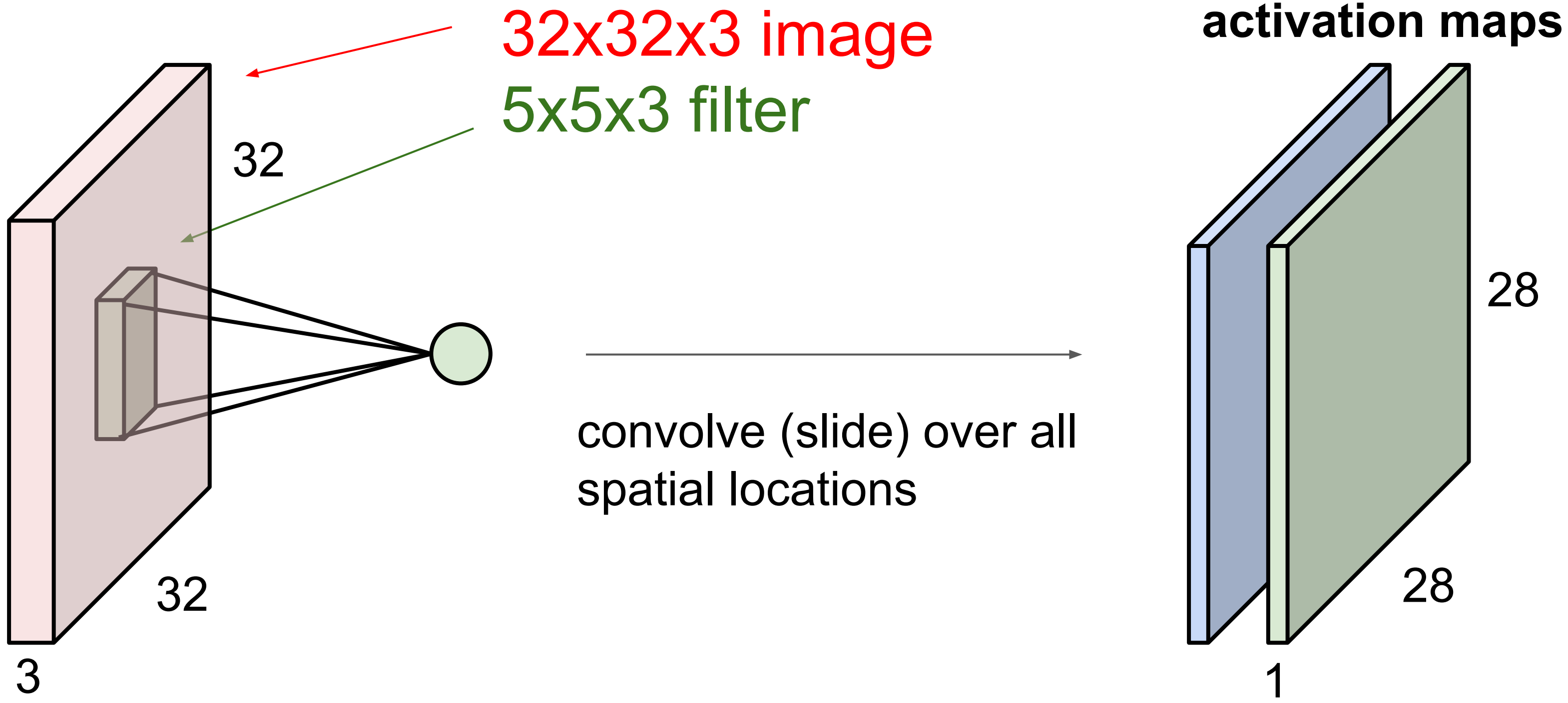
Instance Segmentation

Output:
Category label and instance label for each pixel in the image

Distinguishes between different instances of an object

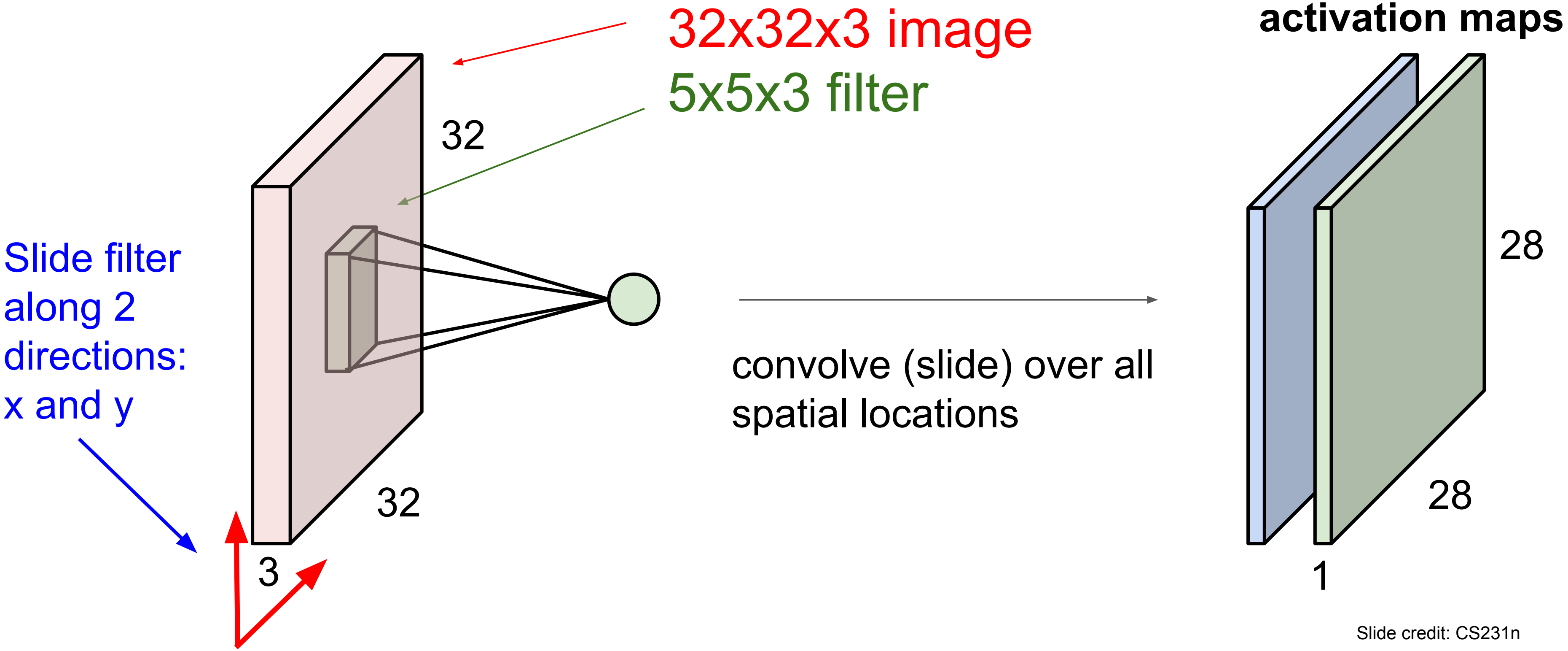
Figures: Chen et al. 2016. https://arxiv.org/pdf/1604.02677.pdf

Also Last Time:
Remember 2D convolutions



Slide credit: CS231n

Also Last Time:
Remember 2D convolutions



Slide credit: CS231n

Also Last Time:
3D convolutions

Slide filter along **3** directions: x, y, and z!

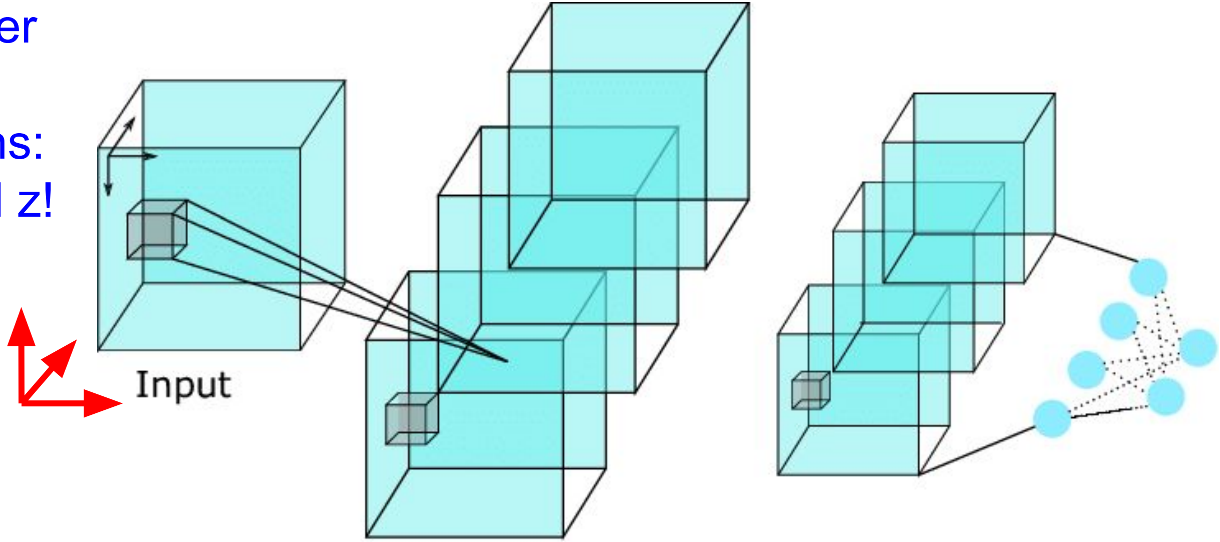
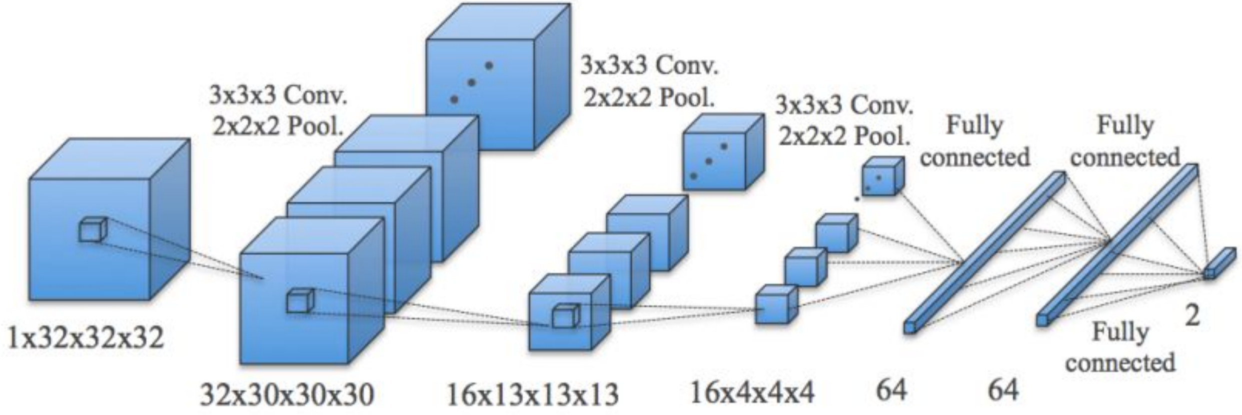


Figure credit:
https://www.researchgate.net/profile/Deepak_Mishra19/publication/330912338/figure/fig1/AS:723363244810254@1549474645742/Basic-3D-CNN-architecture-the-3D-filter-is-convolved-with-the-video-in-three-dimensions.png

Huang et al. 2017

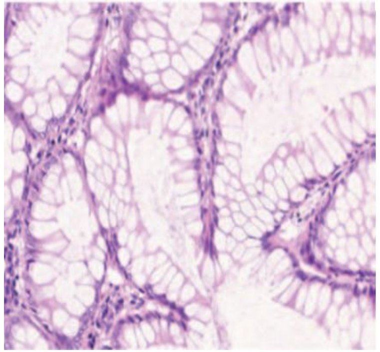
- Simple 3D CNN for lung nodule classification
- Used image processing approaches to extract candidate nodules, then 3D CNN to classify the surrounding volume
- Used the Lung Image Database Consortium (LIDC) Dataset, with 99 3D CT scans



Huang et al. Lung Nodule Detection in CT Using 3D Convolutional Neural Networks. ISBI 2017.

For richer visual recognition tasks, can also extend respective CNN architectures to use 3D convolutions

Classification

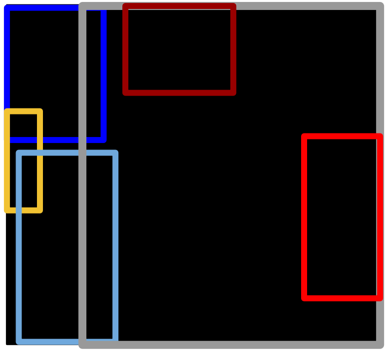


Output:
one category label for image (e.g., colorectal glands)

Semantic Segmentation

Output:
category label for each pixel in the image

Detection



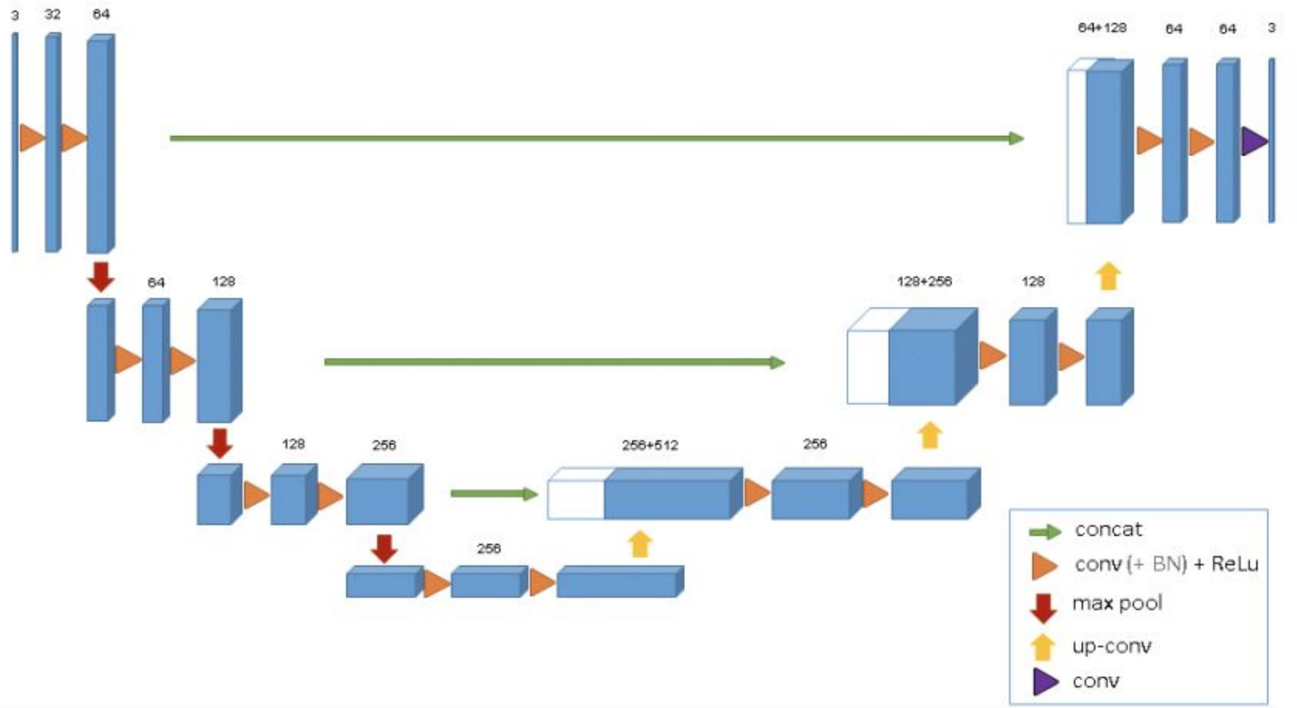
Output:
Spatial bounding box for each **instance** of a category object in the image

Instance Segmentation

Output:
Category label and instance label for each pixel in the image

Figures: Chen et al. 2016. https://arxiv.org/pdf/1604.02677.pdf

E.g. 3D U-Net



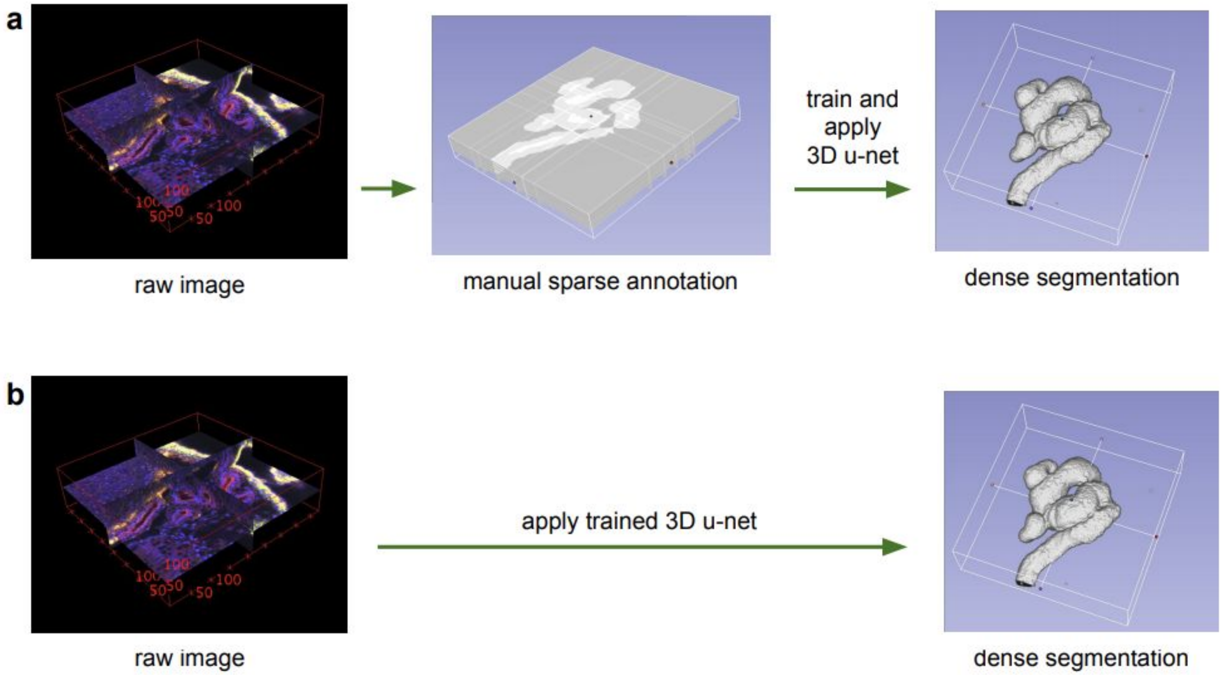
Cicek et al. 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. MICCAI 2016.

E.g. 3D U-Net

Ex: 3D segmentation of Xenopus kidney in confocal microscopic data

Spatial dims: ~ 250 x 250 x 60.
3 channels: each channel corresponds to a different type of data capture

Used only 3 samples total! (with total of 77 annotated 2D slices). Leverages fact that each sample contains many instances of same repetitive structures w/ variation.



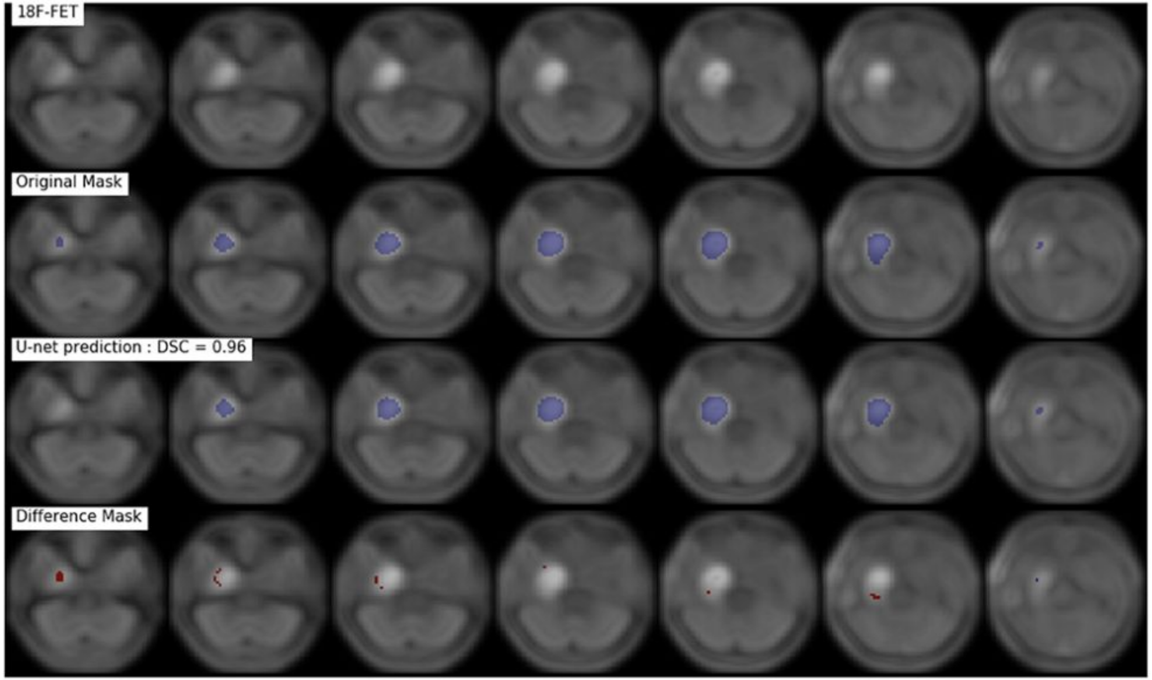
Cicek et al. 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. MICCAI 2016.

Ex: Brain lesion segmentation

Training set: 37 PET scans (3D volumes)

Evaluation set: 11 PET scans

Volumes resized to 64x64x40 for computational efficiency

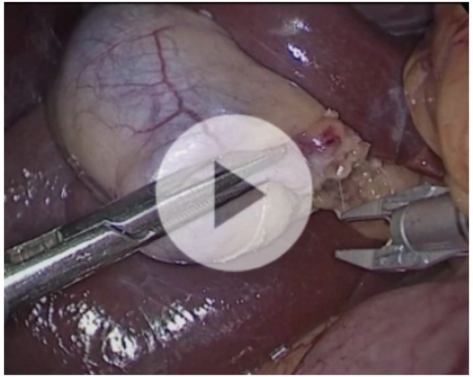


Blanc-Durand et al. Automatic lesion detection and segmentation of 18F-FET PET in gliomas: A full 3D U-Net convolutional neural network study. PLoS One, 2018.

Video data (high dimensional in time)

E.g. in:

Surgery



Hospital patient monitoring

Psychology

3D convolutions

Slide filter along **3** directions: x, y, and z

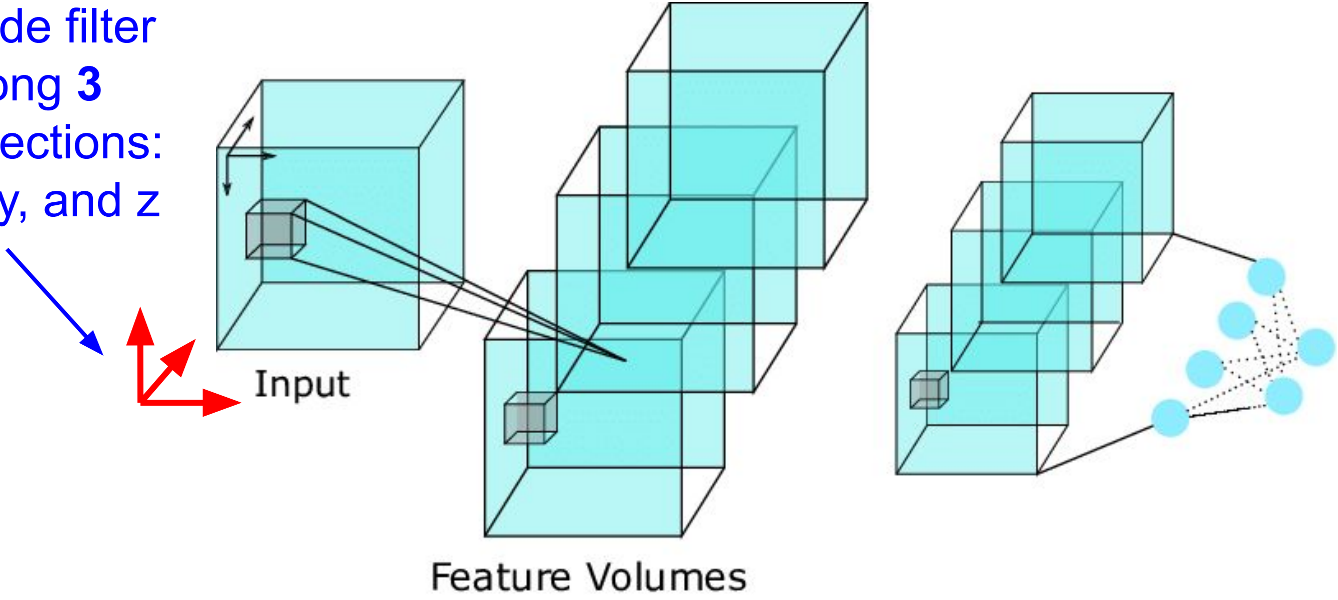
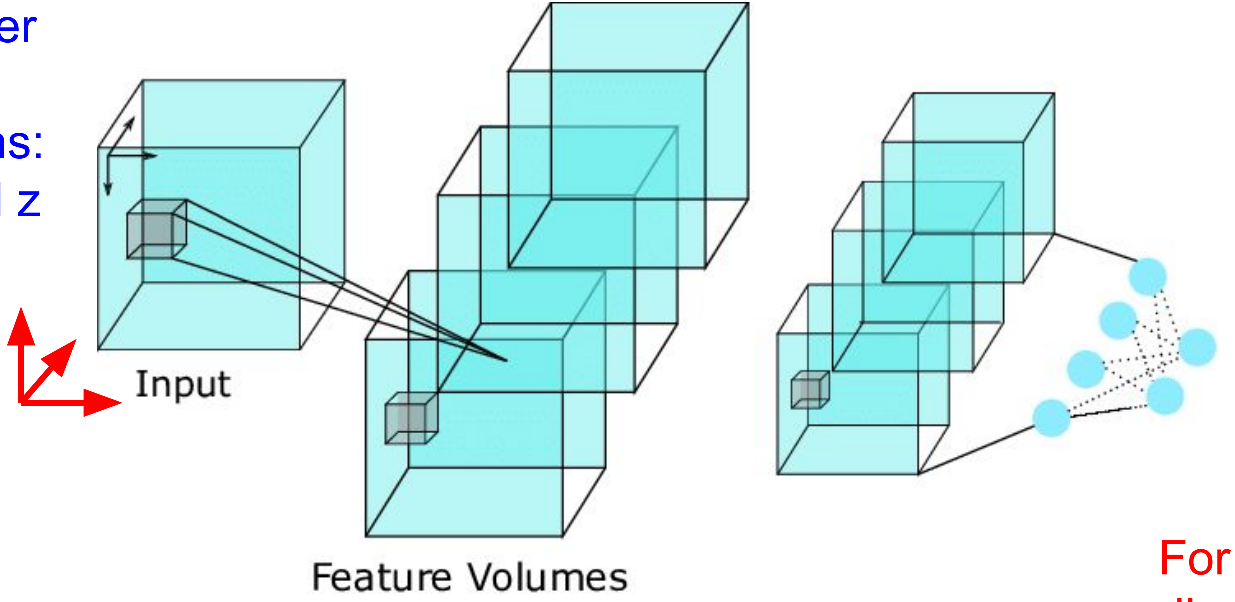


Figure credit:
https://www.researchgate.net/profile/Deepak_Mishra19/publication/330912338/figure/fig1/AS:723363244810254@1549474645742/Basic-3D-CNN-architecture-the-3D-filter-is-convolved-with-the-video-in-three-dimensions.png

3D convolutions

Slide filter along **3** directions: x, y, and z

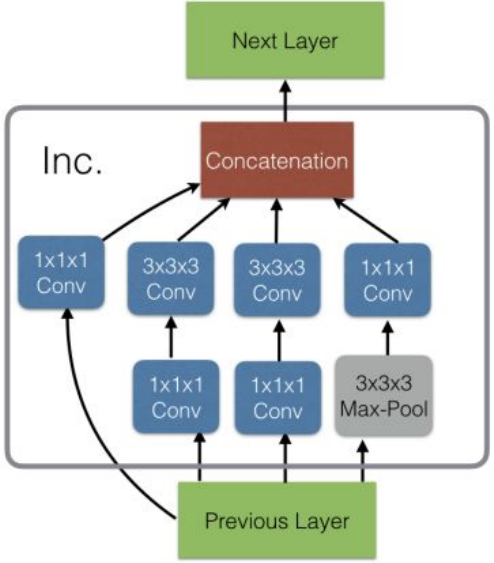


For video data, 3rd dimension is time

Figure credit:
https://www.researchgate.net/profile/Deepak_Mishra19/publication/330912338/figure/fig1/AS:723363244810254@1549474645742/Basic-3D-CNN-architecture-the-3D-filter-is-convolved-with-the-video-in-three-dimensions.png

I3D: 3D convolutional network for video data

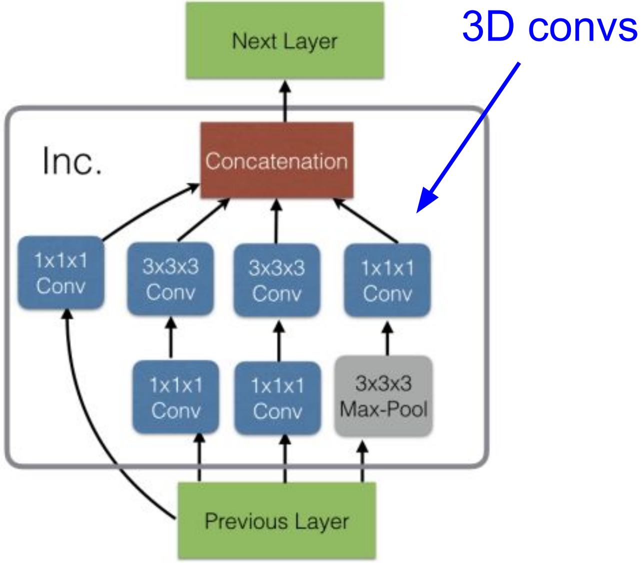
Inception Module (Inc.) w/
3D convolutions



Carreira and Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. CVPR 2017.

I3D: 3D convolutional network for video data

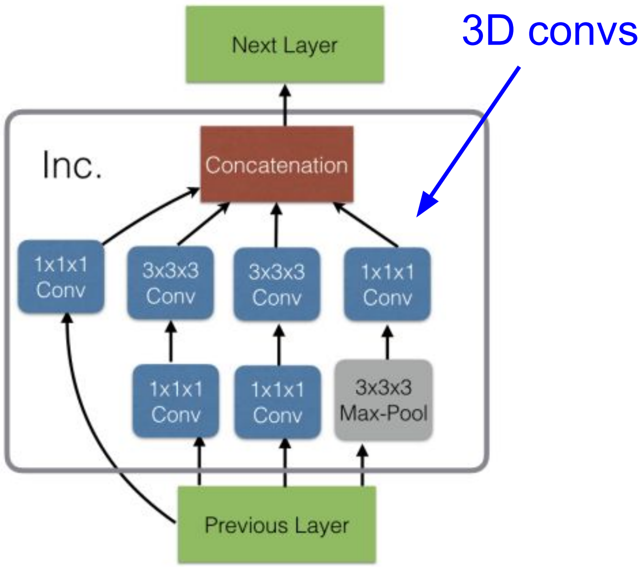
Inception Module (Inc.) w/
3D convolutions



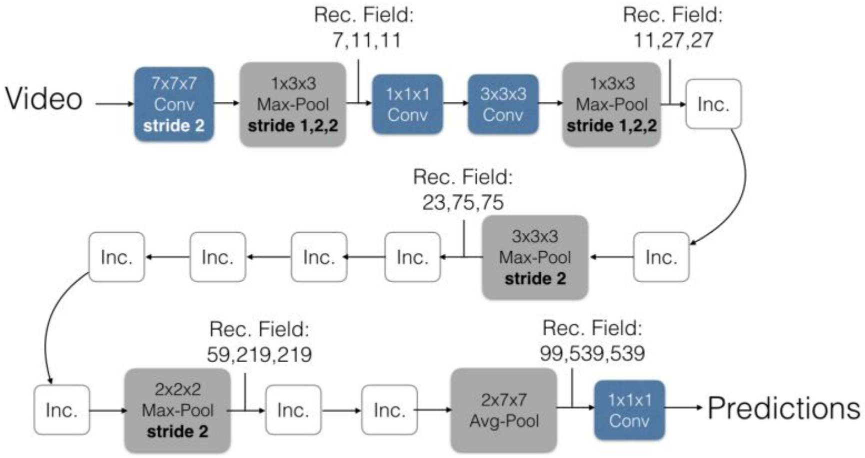
Carreira and Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. CVPR 2017.

I3D: 3D convolutional network for video data

Inception Module (Inc.) w/
3D convolutions



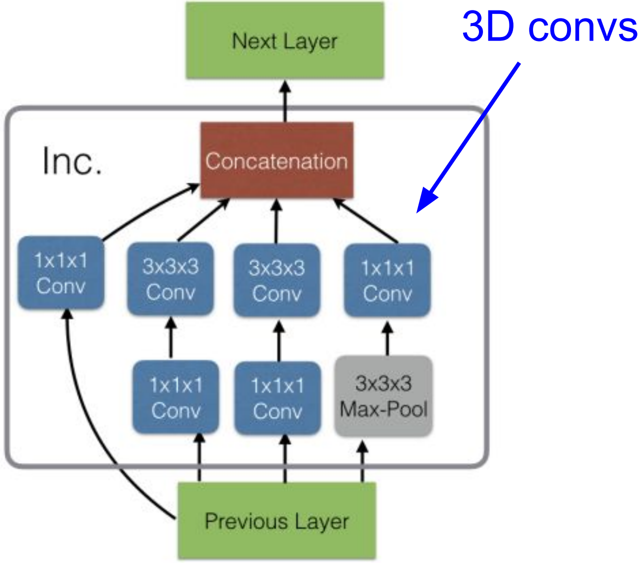
3D Inception Module used in Inception
Network (also known as GoogLeNet)



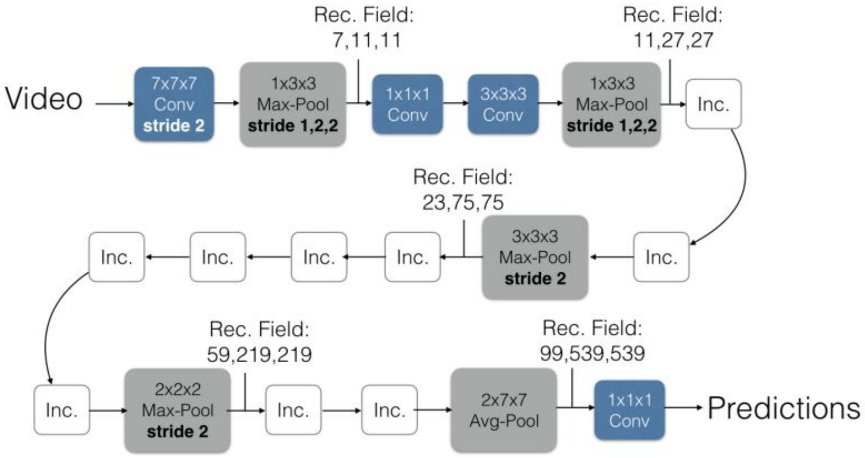
Carreira and Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. CVPR 2017.

I3D: 3D convolutional network for video data

Inception Module (Inc.) w/ 3D convolutions



3D Inception Module used in Inception Network (also known as GoogLeNet)



Can pre-train from 2D datasets e.g. ImageNet by replicating and normalizing 2D weights over additional dimension!

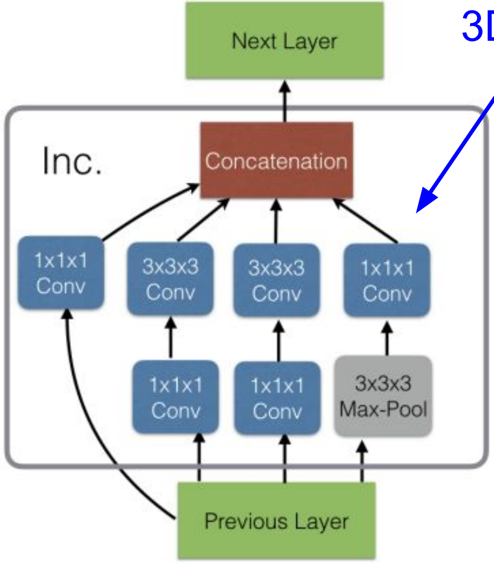
Carreira and Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. CVPR 2017.

I3D: 3D convolutional network for video data

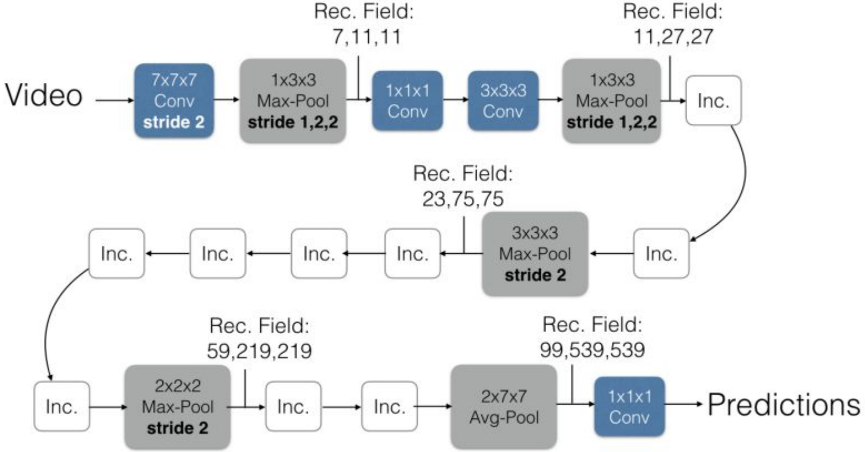
Note: in general, can 3D-ify many 2D architectures!

Inception Module (Inc.) w/ 3D convolutions

3D convs



3D Inception Module used in Inception Network (also known as GoogLeNet)



Can pre-train from 2D datasets e.g. ImageNet by replicating and normalizing 2D weights over additional dimension!

Carreira and Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. CVPR 2017.

Video classifiers (including I3D) often enhanced with optical flow

Two consecutive frames

Optical flow displacement vectors

horizontal (L) and vertical (R) components of displacement

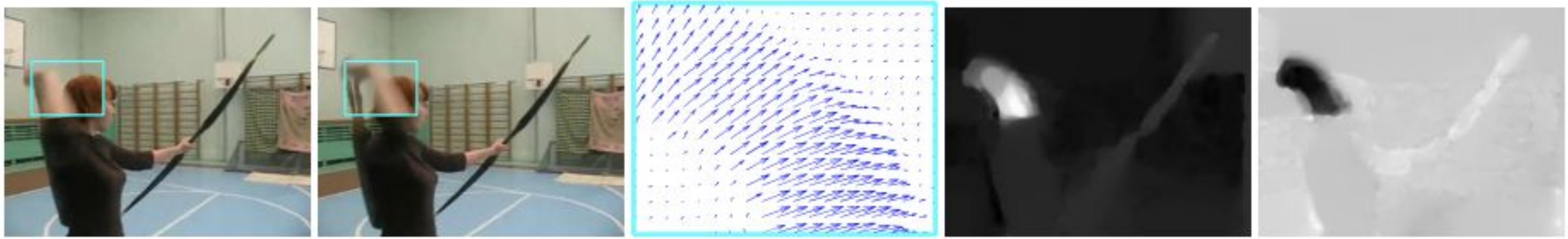
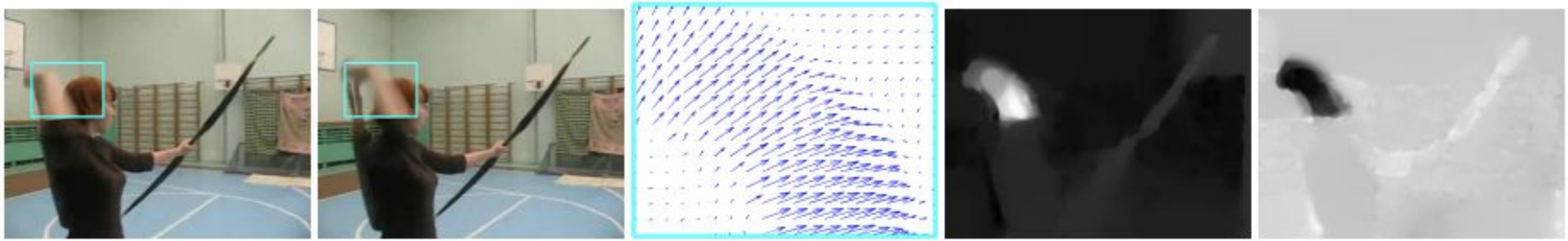


Figure credit: Simonyan and Zisserman. Two-Stream Convolutional Networks for Action Recognition in Videos. NeurIPS 2014.

Video classifiers (including I3D) often enhanced with optical flow



Two consecutive frames

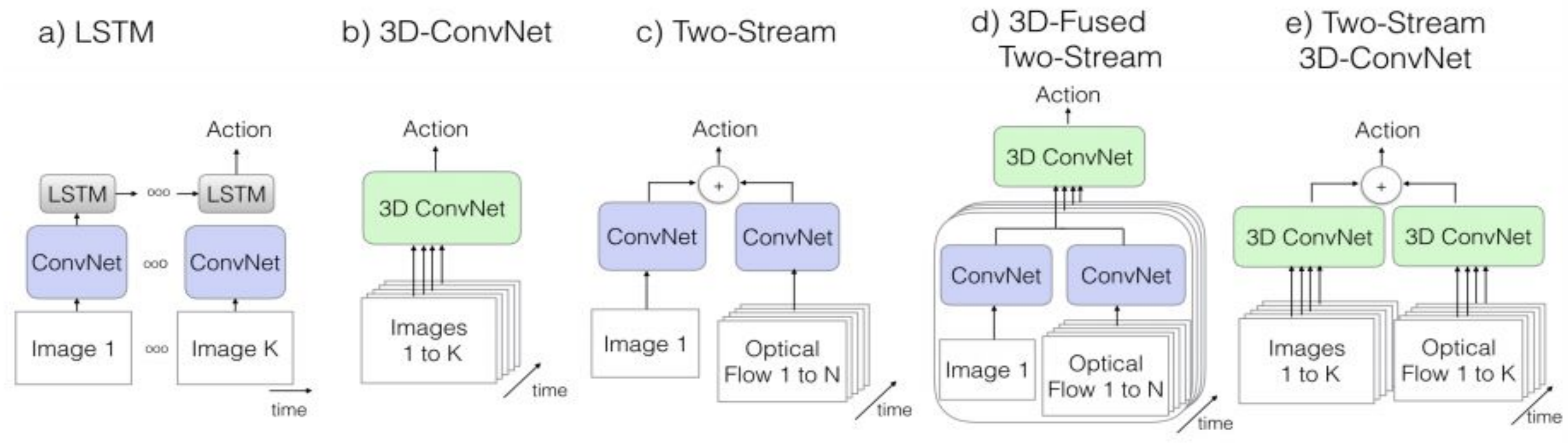
Optical flow displacement vectors

horizontal (L) and vertical (R) components of displacement

Directional components can be represented as images (or multiple channels of input volume!)

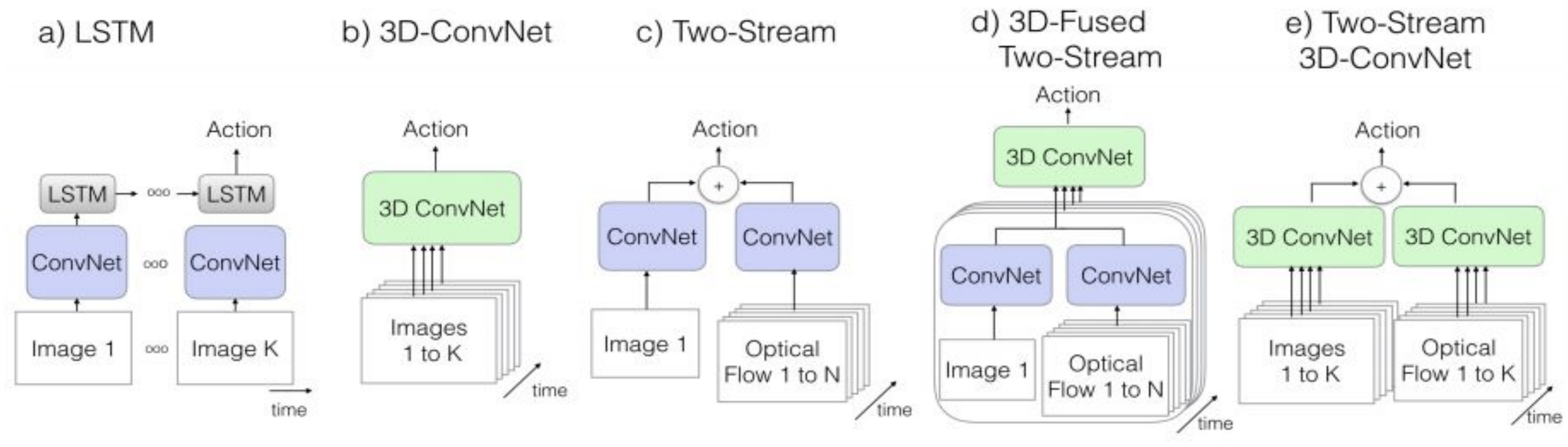
Figure credit: Simonyan and Zisserman. Two-Stream Convolutional Networks for Action Recognition in Videos. NeurIPS 2014.

Video classifiers (including I3D) often enhanced with optical flow



Carreira and Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. CVPR 2017.

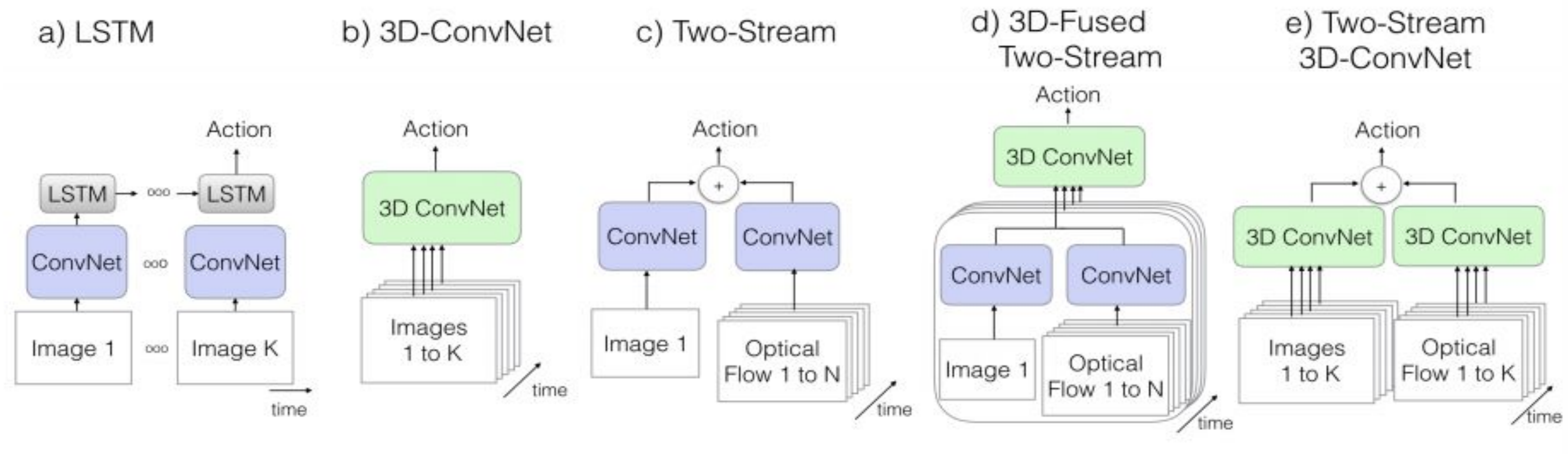
Video classifiers (including I3D) often enhanced with optical flow



LSTM over RGB

Carreira and Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. CVPR 2017.

Video classifiers (including I3D) often enhanced with optical flow

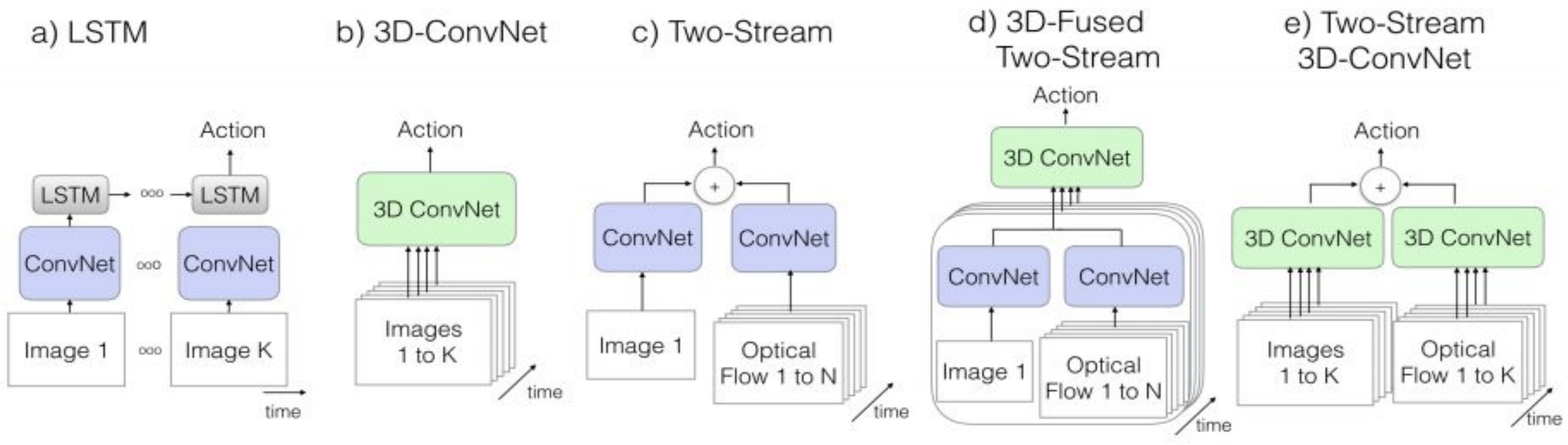


LSTM over RGB

(LSTM is a type of recurrent neural network.
We will talk more about these soon!)

Carreira and Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. CVPR 2017.

Video classifiers (including I3D) often enhanced with optical flow

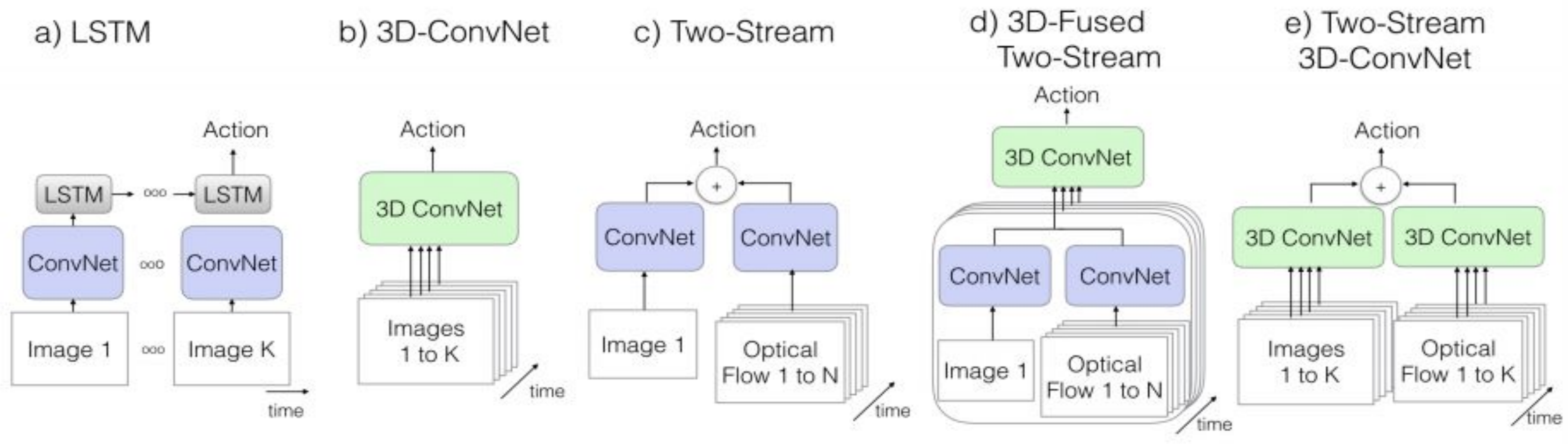


LSTM over RGB

I3D (3D convs) over RGB

Carreira and Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. CVPR 2017.

Video classifiers (including I3D) often enhanced with optical flow



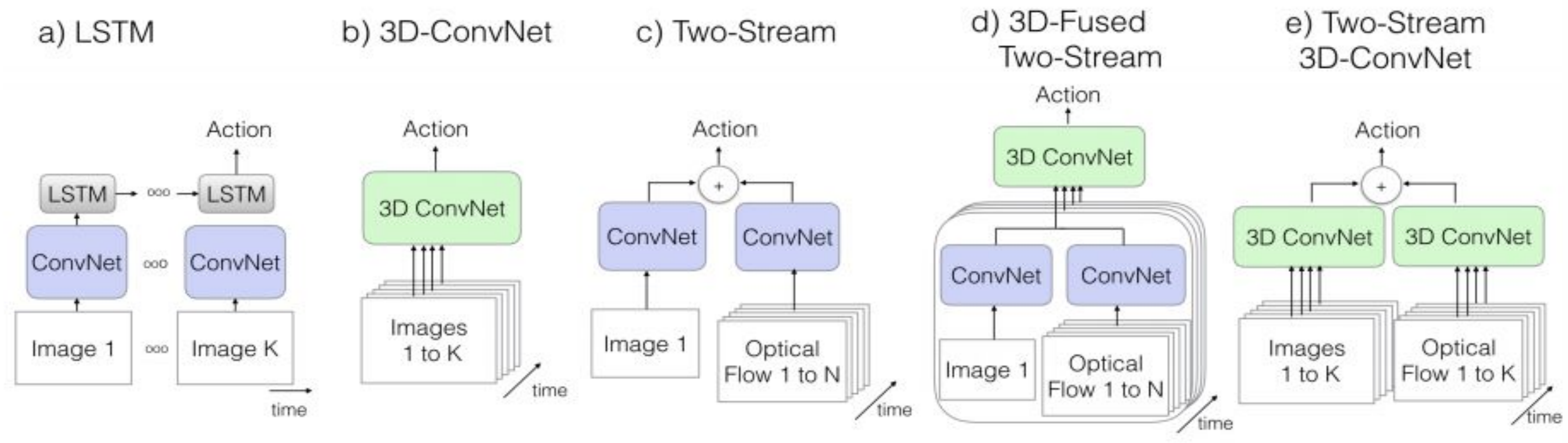
LSTM over RGB

I3D (3D convs) over RGB

2D convs over RGB + optical flow (OF)

Carreira and Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. CVPR 2017.

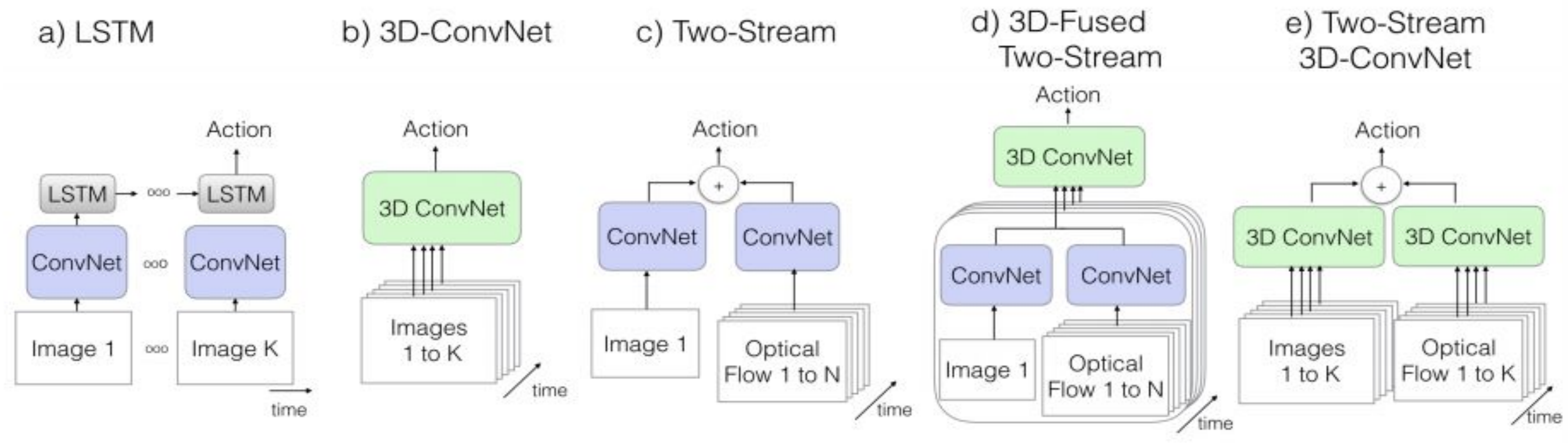
Video classifiers (including I3D) often enhanced with optical flow



LSTM over RGB | I3D (3D convs) over RGB | 2D convs over RGB + optical flow (OF) | Late 3D fusion of RGB + OF

Carreira and Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. CVPR 2017.

Video classifiers (including I3D) often enhanced with optical flow



LSTM over RGB

I3D (3D convs) over RGB

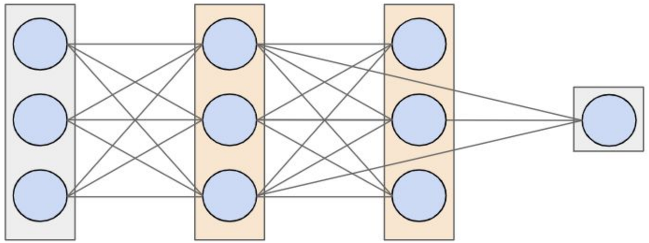
2D convs over RGB + optical flow (OF)

Late 3D fusion of RGB + OF

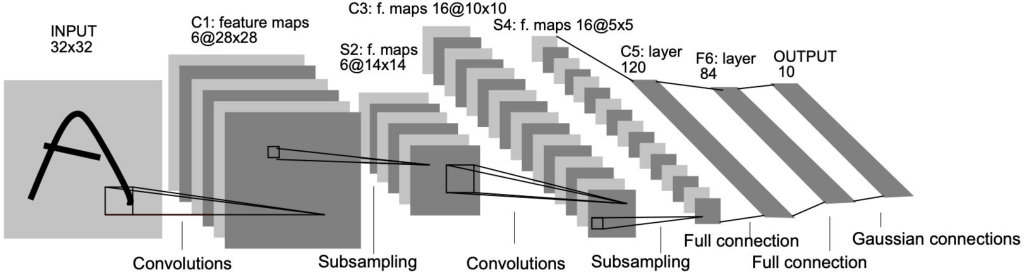
Two I3D streams over RGB + OF

Carreira and Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. CVPR 2017.

Preview: Recurrent neural networks

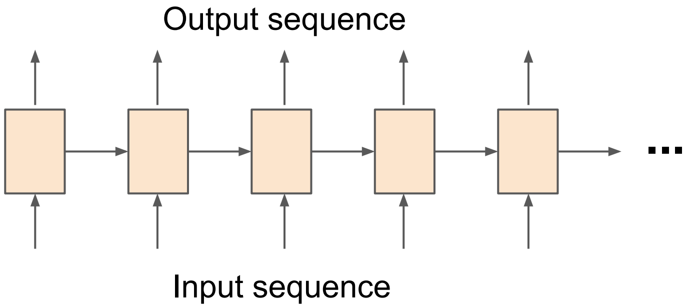


Fully connected neural networks
(linear layers, good for “feature vector” inputs)



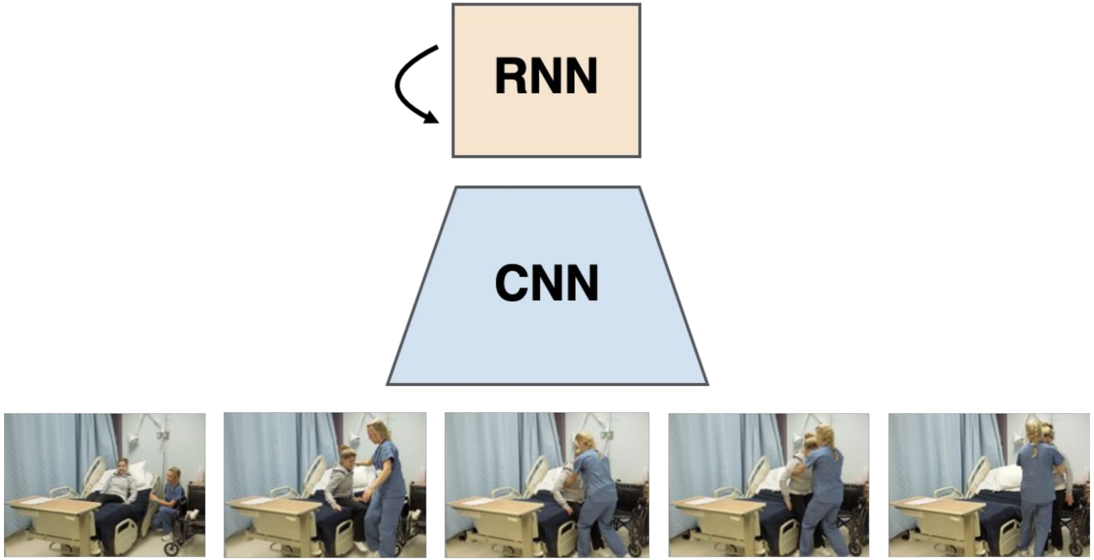
Convolutional neural networks
(convolutional layers, good for image inputs)

Recurrent neural networks
(linear layers modeling recurrence relation across sequence, good for sequence inputs)



Videos are sequences: natural fit for recurrent networks

$$\mathbf{y} = \{y_0, y_1, ..., y_T\}$$

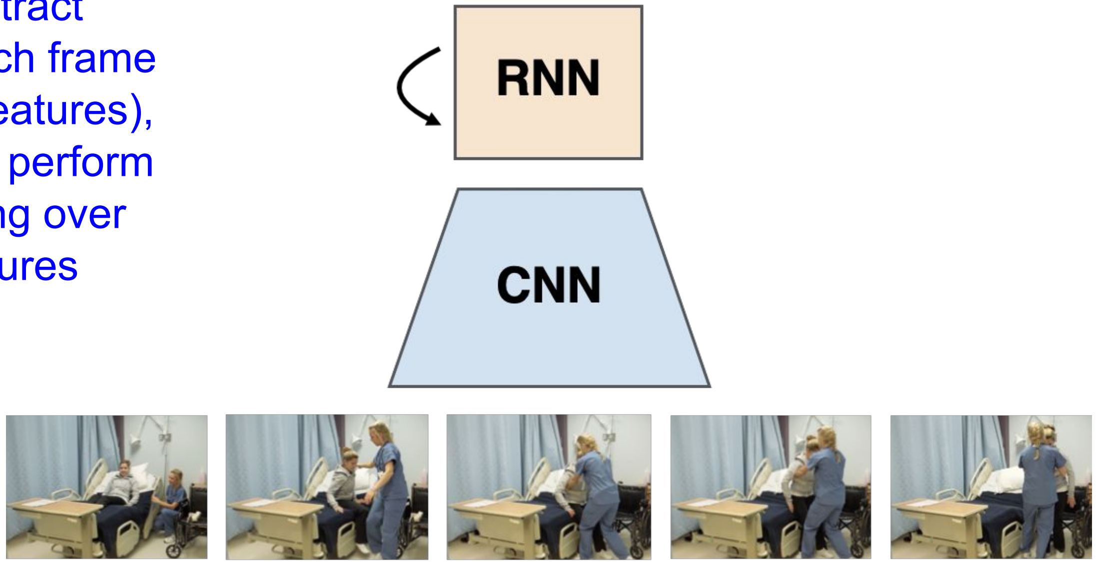


$$\mathbf{x} = \{x_0, x_1, ..., x_T\}$$

Videos are sequences: natural fit for recurrent networks

Abstracted overview:
Use a CNN to extract features from each frame (e.g. final-layer features), then use RNN to perform temporal modeling over sequence of features

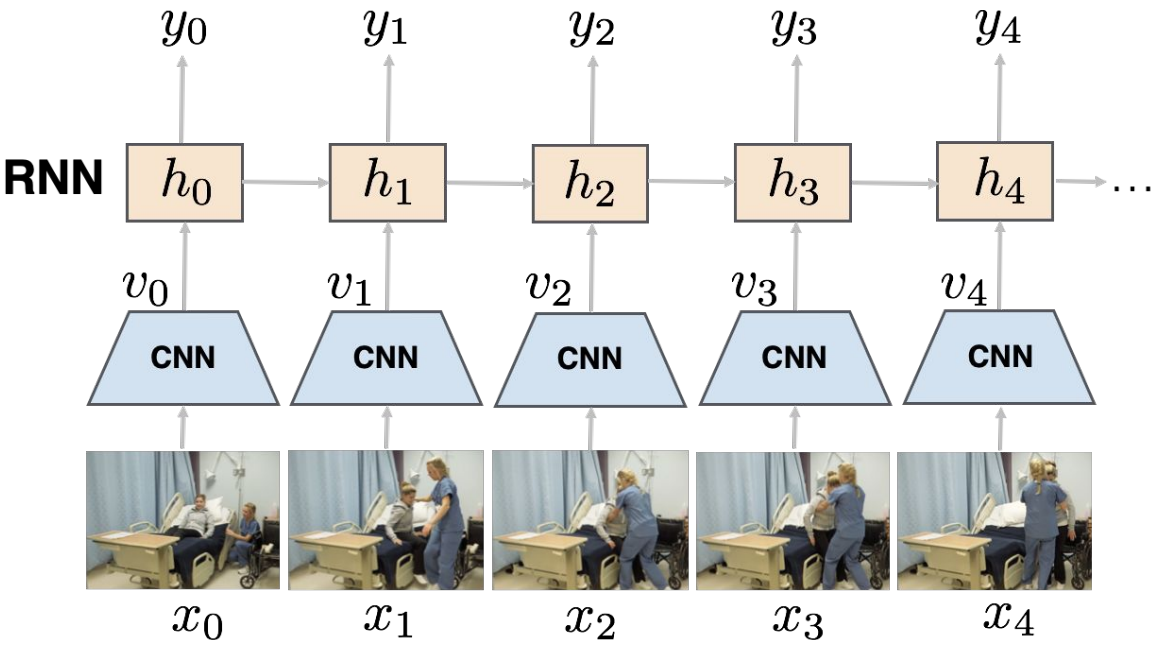
$$\mathbf{y} = \{y_0, y_1, ..., y_T\}$$



$$\mathbf{x} = \{x_0, x_1, ..., x_T\}$$

Videos are sequences: natural fit for recurrent networks

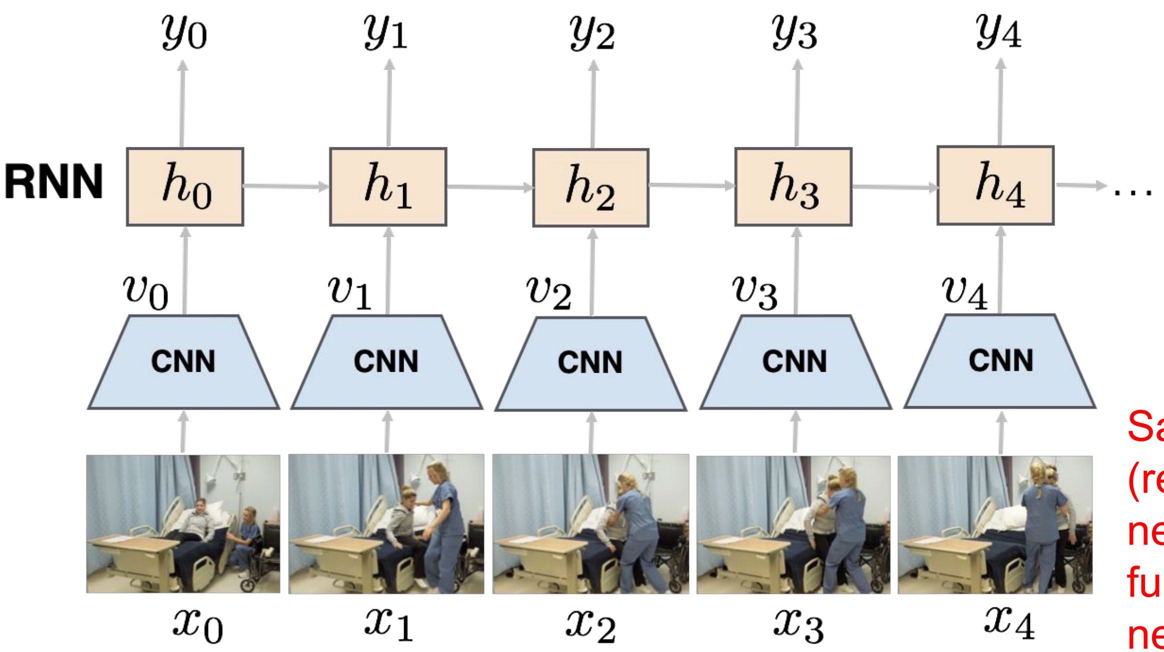
Diagram of a CNN + RNN “rolled out” over time



$$
\begin{aligned}
h_t &= f_W(h_{t-1}, v_t) \\
&= \tanh(W_{hh}h_{t-1} + W_{vh}v_t) \\
y_t &= W_{hy}h_t
\end{aligned}
$$

Videos are sequences: natural fit for recurrent networks

Diagram of a CNN + RNN “rolled out” over time



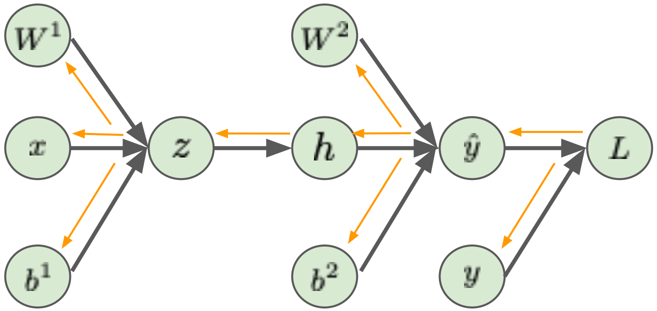
$$
\begin{aligned}
h_t &= f_W(h_{t-1}, v_t) \\
&= \boxed{\tanh(W_{hh}h_{t-1} + W_{vh}v_t)} \\
y_t &= \boxed{W_{hy}h_t}
\end{aligned}
$$

Same idea of weight matrices (remember fully-connected networks) and nonlinear activation functions! Just applied to a neural network with a different connectivity structure

Aside: how do we compute gradient updates? Remember backpropagation.

Network output: $\hat{y} = W^2(\sigma(W^1 x + b^1)) + b^2$

Think of computing loss function as staged computation of intermediate variables:



"Forward pass":

$$z = W^1 x + b^1$$
$$h = \sigma(z)$$
$$\hat{y} = W^2 h + b^2$$
$$L = (\hat{y} - y)^2$$

Now, can use a repeated application of the chain rule, going backwards through the computational graph, to obtain the gradient of the loss with respect to each node of the computation graph.

"Backward pass":

$$\frac{\partial L}{\partial \hat{y}} = 2(\hat{y} - y)$$

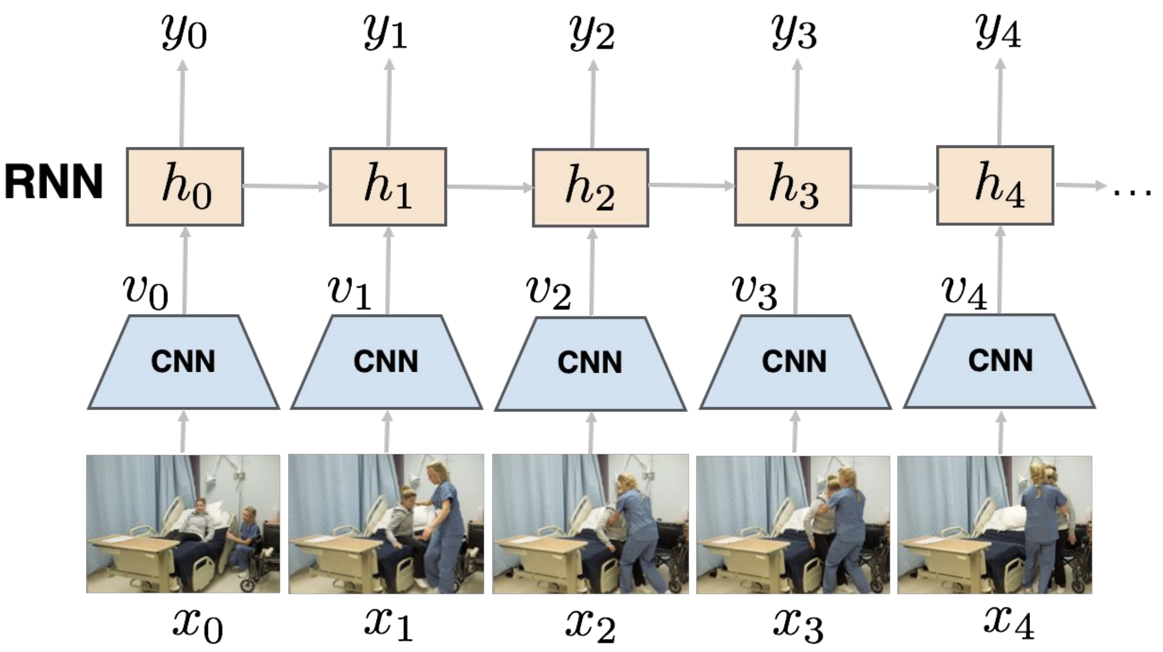
(not all gradients shown)

$$\frac{\partial L}{\partial W^2} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial W^2}$$
$$\frac{\partial L}{\partial H} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial H}$$
$$\frac{\partial L}{\partial Z} = \frac{\partial L}{\partial H} \frac{\partial H}{\partial Z}$$
$$\frac{\partial L}{\partial W^1} = \frac{\partial L}{\partial Z} \frac{\partial z}{\partial W^1}$$

Plug in from earlier computations via chain rule

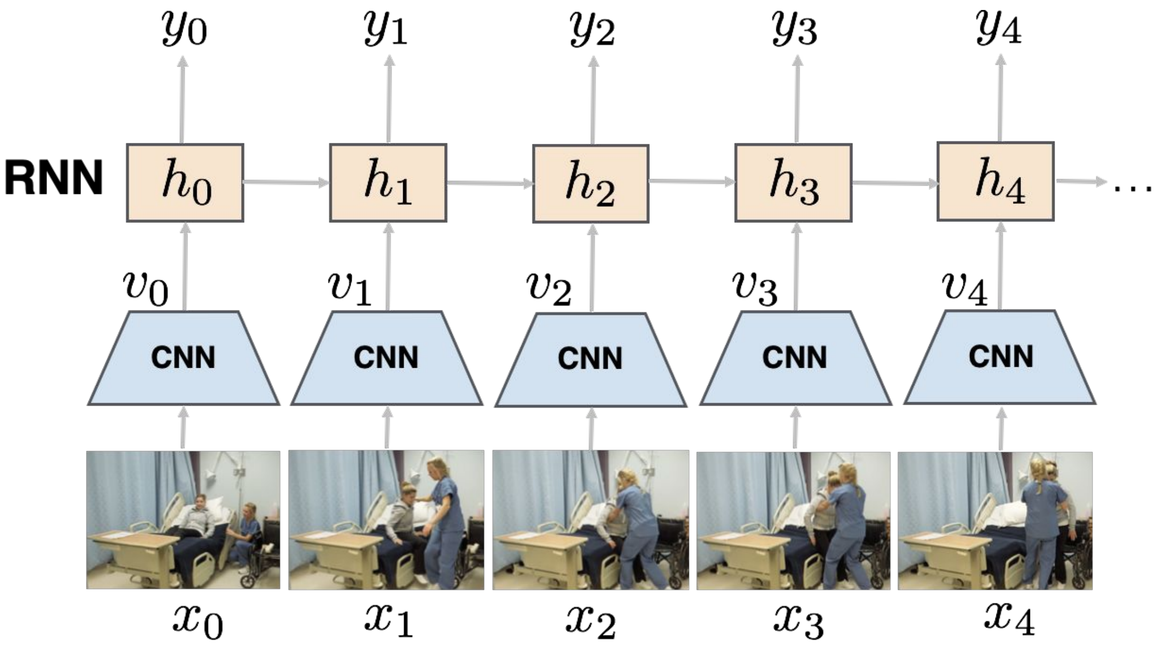
Local gradients to derive

Videos are sequences: natural fit for recurrent networks



This is a computational graph
-> can backprop and train
RNN and CNN jointly

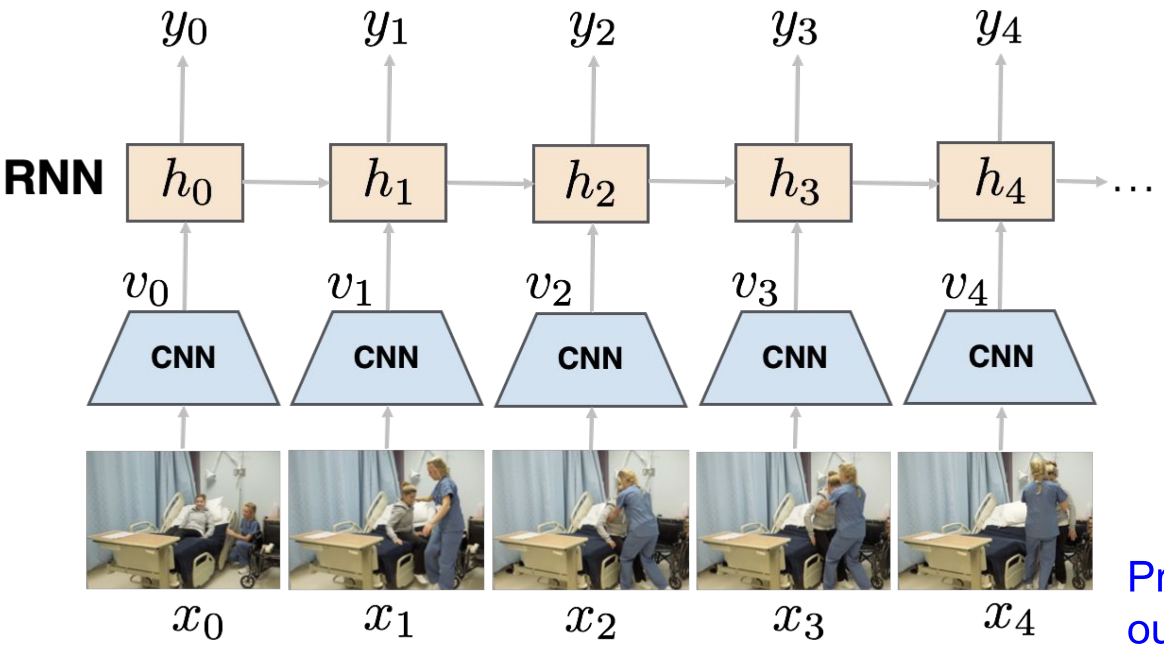
Videos are sequences: natural fit for recurrent networks



This is a computational graph -> can backprop and train RNN and CNN jointly

But a very large number of parameters to train simultaneously… more common to fine-tune a single-frame CNN over the data first (or use pre-trained CNN), then extract features and train the RNN separately

Videos are sequences: natural fit for recurrent networks

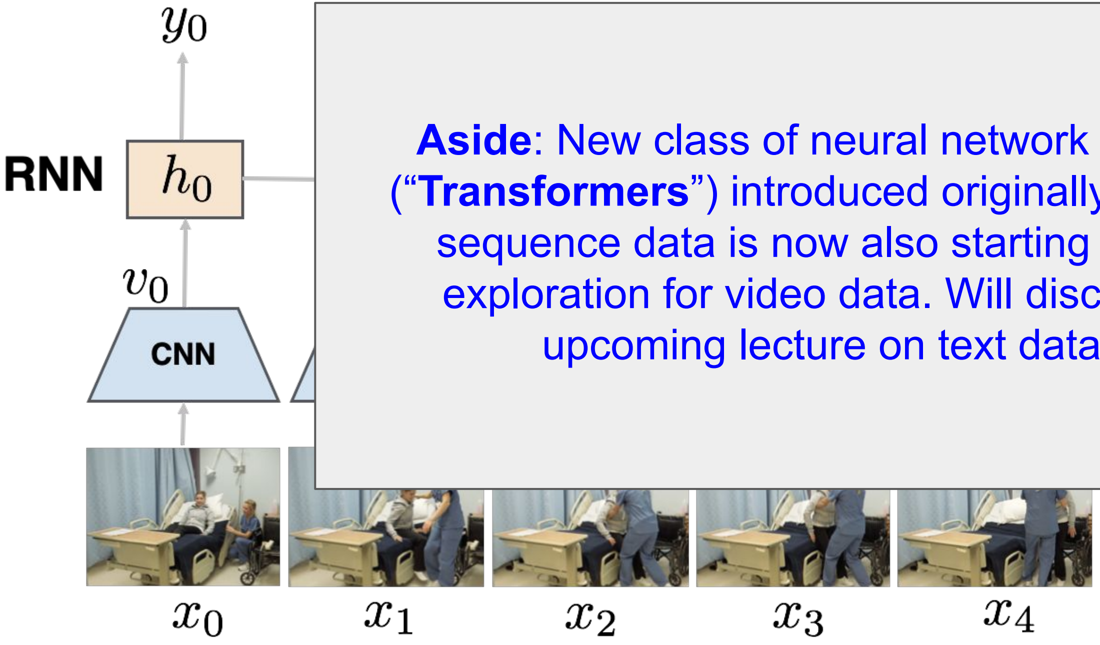


This is a computational graph -> can backprop and train RNN and CNN jointly

But a very large number of parameters to train simultaneously… more common to fine-tune a single-frame CNN over the data first (or use pre-trained CNN), then extract features and train the RNN separately

Preview of RNNs. Will see again in our discussion of sequence EHR data.

Videos are sequences: natural fit for recurrent networks



Aside: New class of neural network models ("**Transformers**") introduced originally for NLP sequence data is now also starting to see exploration for video data. Will discuss in upcoming lecture on text data.

This is a computational graph -> can backprop and train ... d CNN jointly

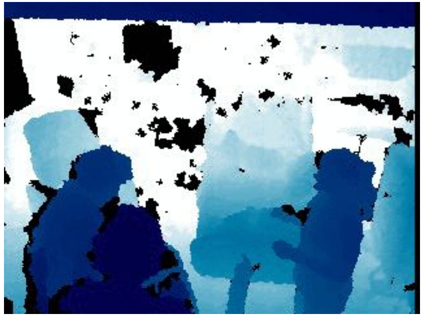
...ery large number of ...ters to train ...neously… more ...n to fine-tune a ...ame CNN over the ...t (or use pre-trained ...hen extract features and train the RNN separately

Preview of RNNs. Will see again in our discussion of sequence EHR data.

Detecting patient mobilization activities in the ICU

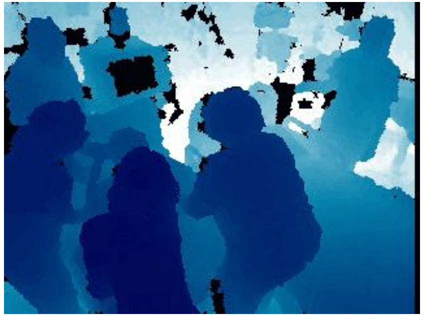
Get patient out of bed

Sit patient in chair

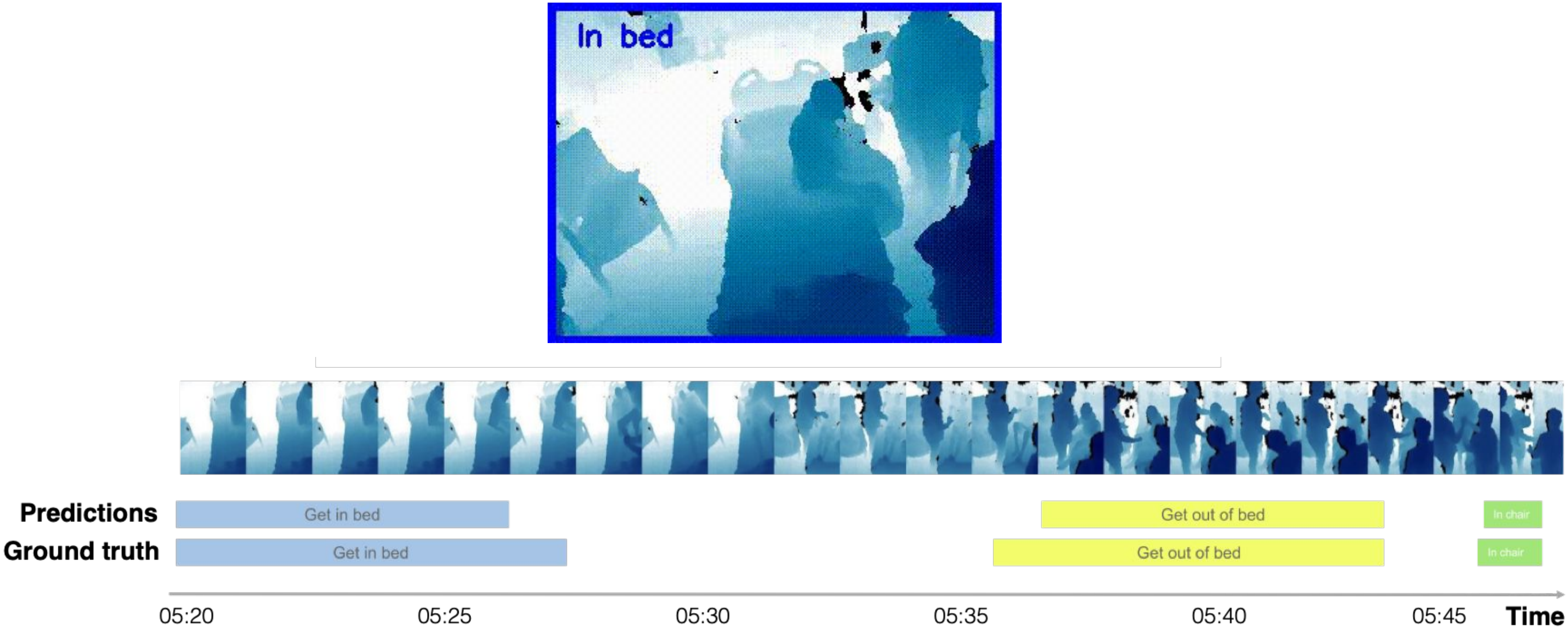


Get patient in bed

Get patient out of chair

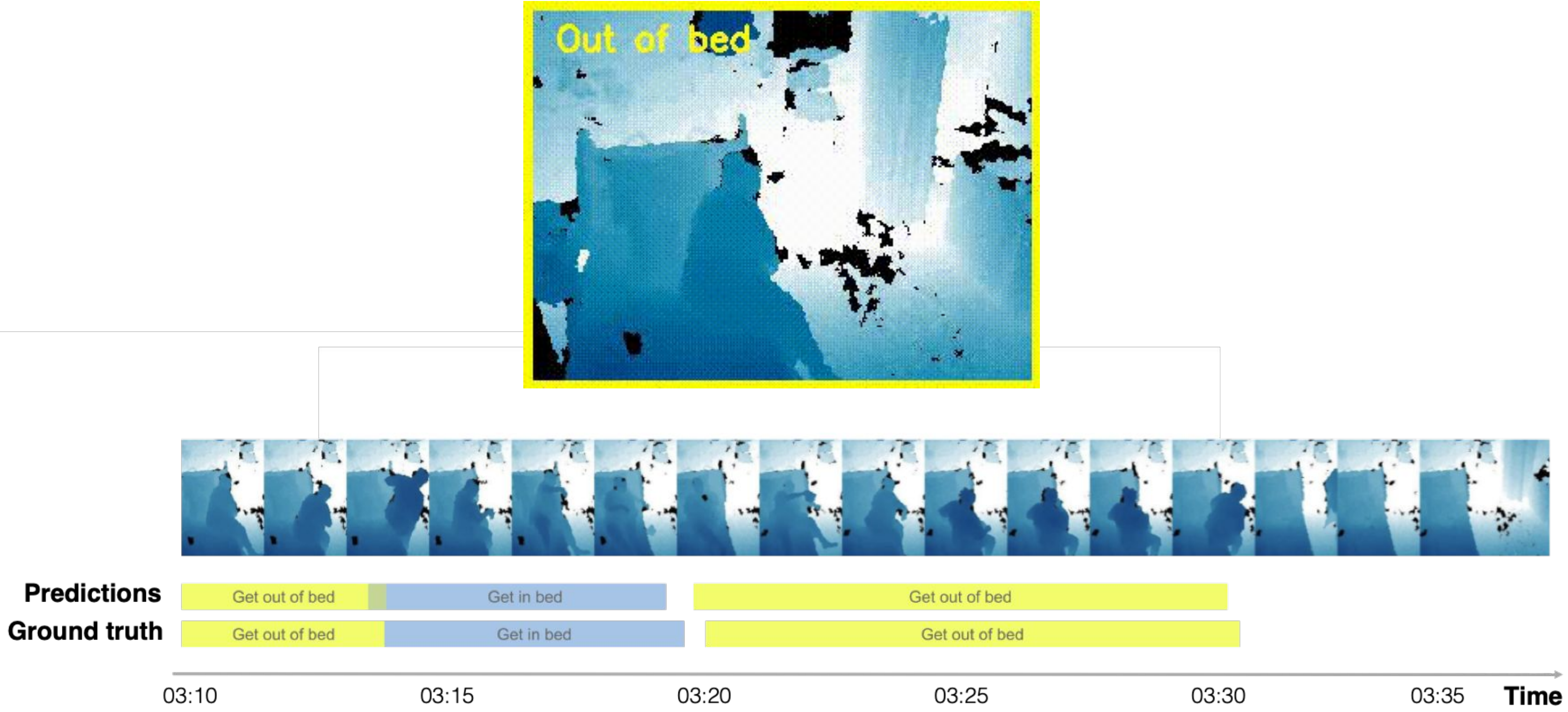


Detecting patient mobilization activities in the ICU



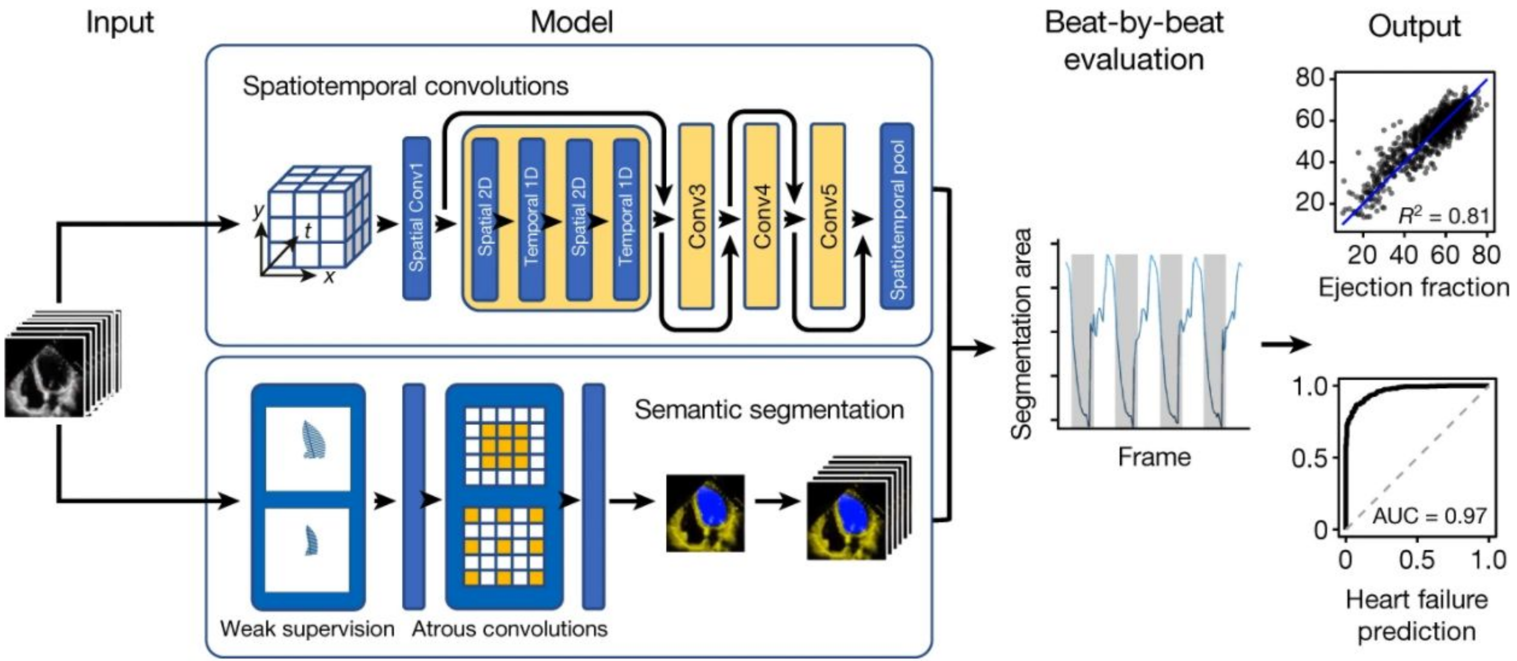
Yeung*,Salipur*, et al. A Computer Vision System for Deep Learning-Based Detection of Patient Mobilization Activities in the ICU. npj Digital Medicine, 2019.

Detecting patient mobilization activities in the ICU



Yeung*,Salipur*, et al. A Computer Vision System for Deep Learning-Based Detection of Patient Mobilization Activities in the ICU. npj Digital Medicine, 2019.

Predicting ejection fraction in echocardiograms



Ouyang et al. Video-based AI for beat-to-beat assessment of cardiac function. Nature, 2020.

Summary

Finished up advanced deep learning models for visual recognition tasks

- Classification
- Semantic segmentation
- **Object detection**
- **Instance segmentation**
- **3D and Video**

Will revisit some of these later with multimodal models and weakly / self- / un-supervised paradigms

Next topic: Introduction to Electronic Health Records

Next Topic:
Electronic Health Records

What are electronic health records?

What are electronic health records?

Patient chart in digital form, containing medical and treatment history

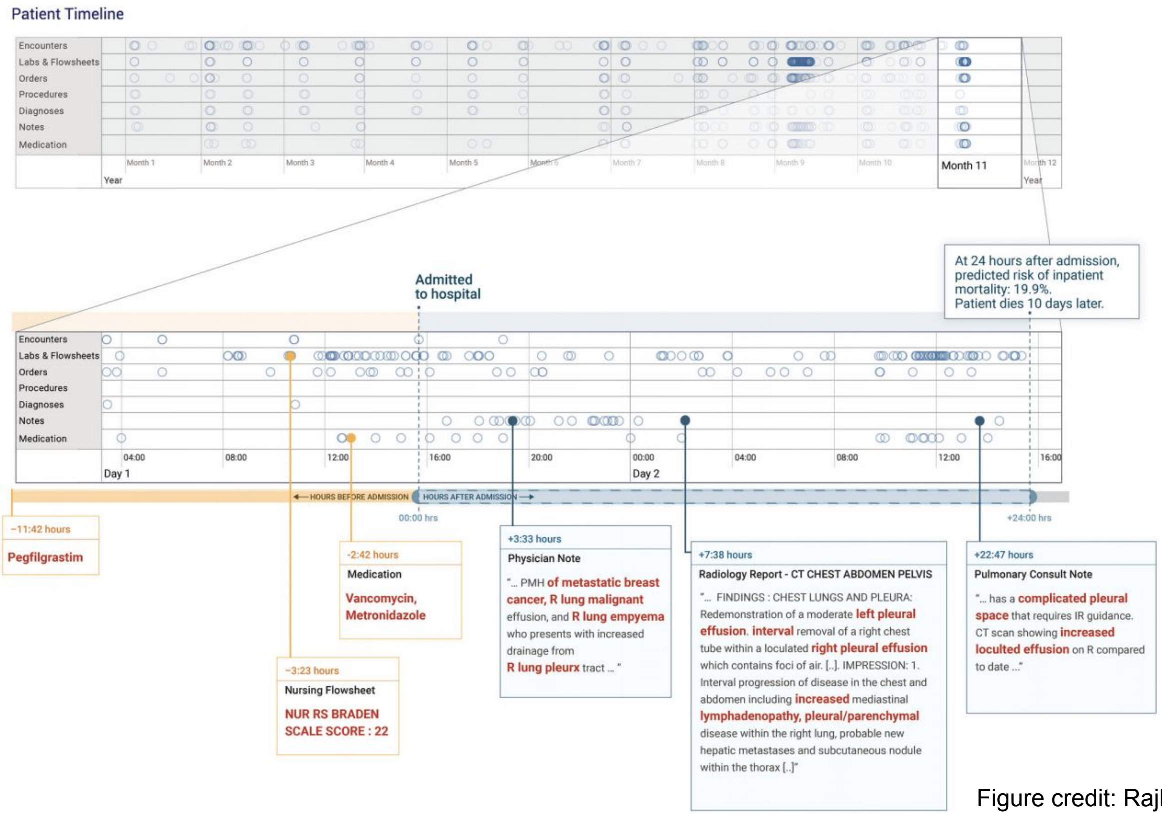


Figure credit: Rajkomar et al. 2018

What are electronic health records?

Patient chart in digital form, containing medical and treatment history

Stores patient information over time

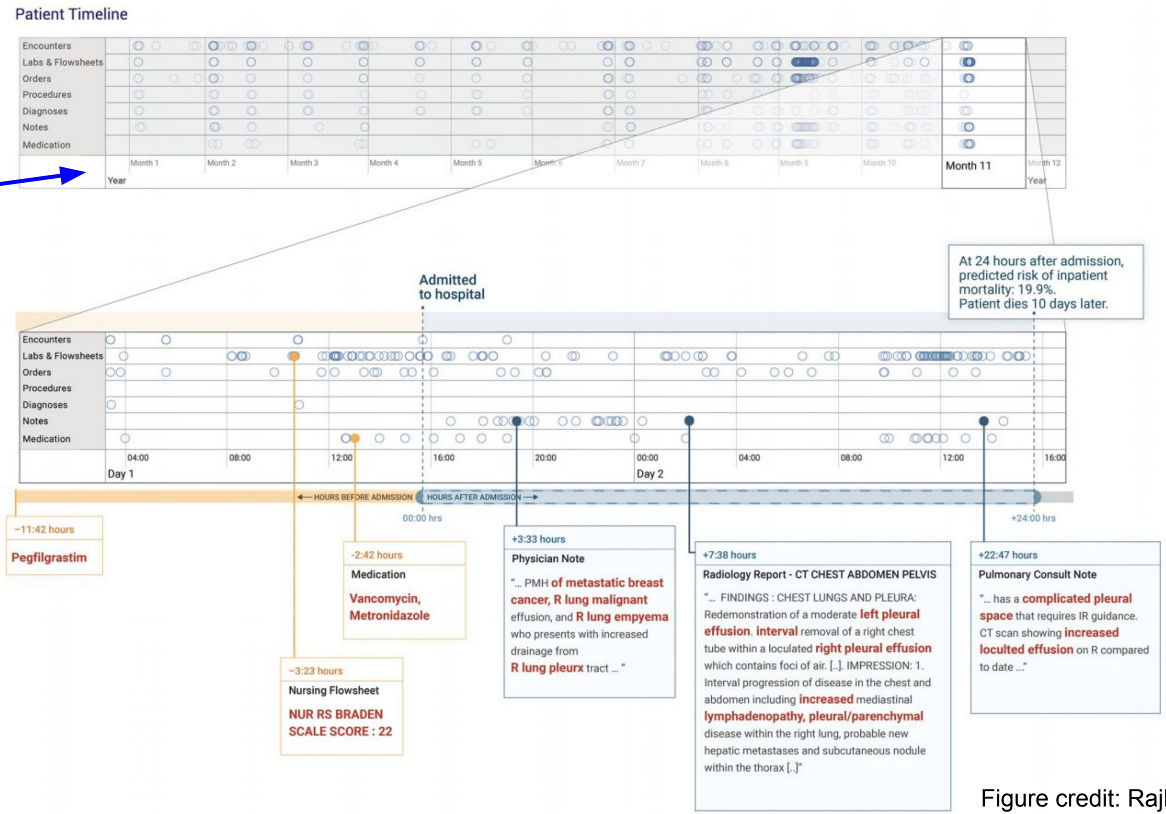
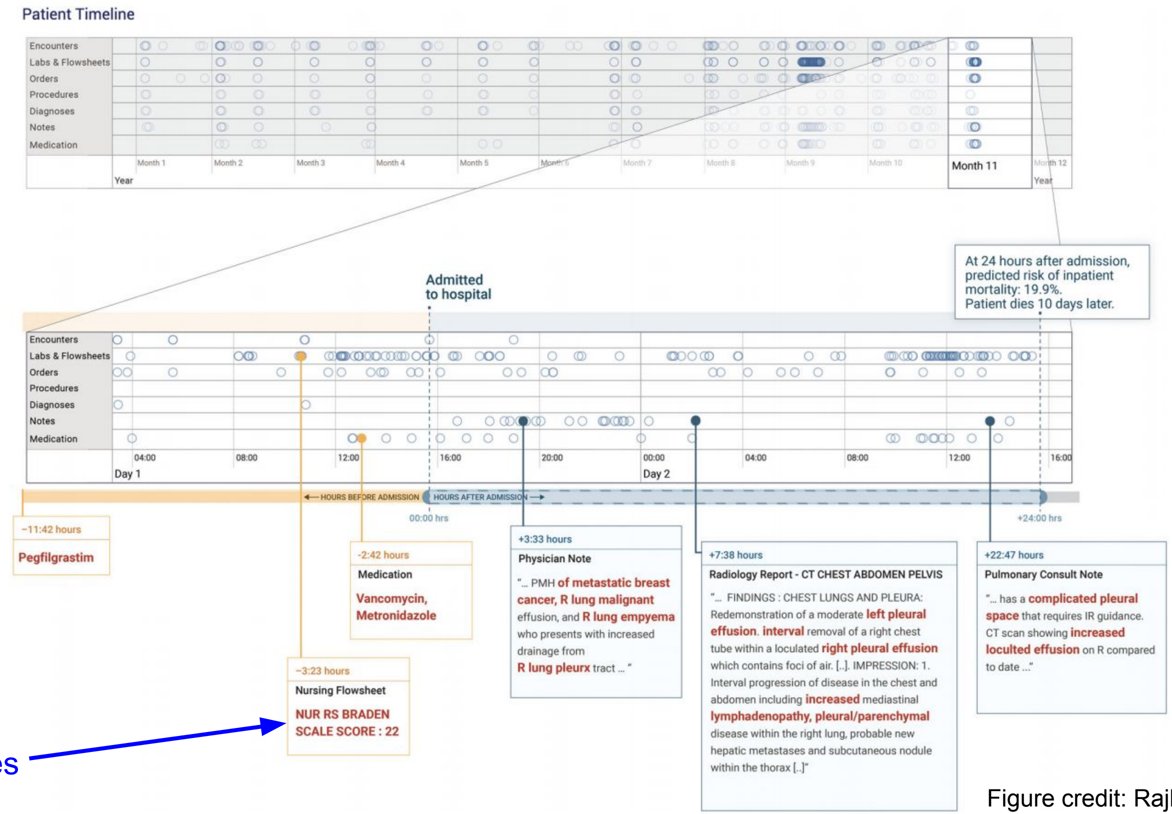


Figure credit: Rajkomar et al. 2018

What are electronic health records?

Patient chart in digital form, containing medical and treatment history

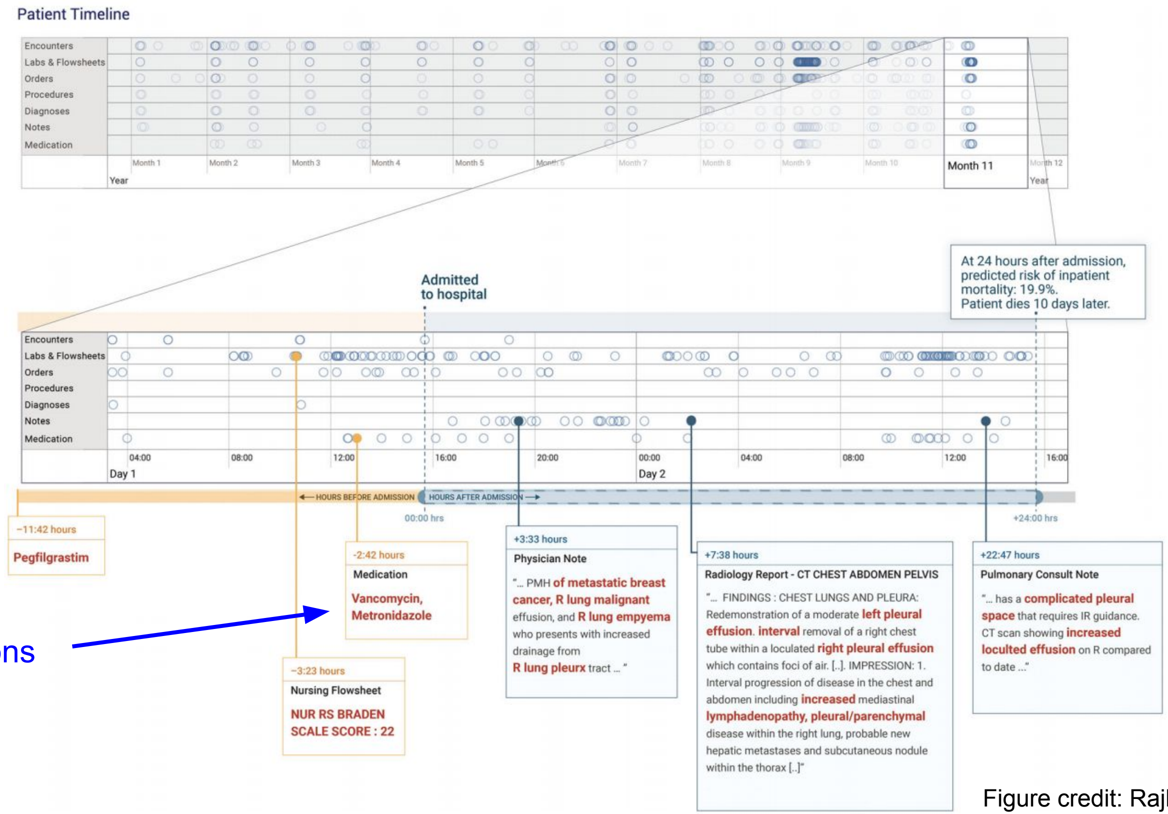


Nursing notes

Figure credit: Rajkomar et al. 2018

What are electronic health records?

Patient chart in digital form, containing medical and treatment history



Medications

Figure credit: Rajkomar et al. 2018

What are electronic health records?

Patient chart in digital form, containing medical and treatment history

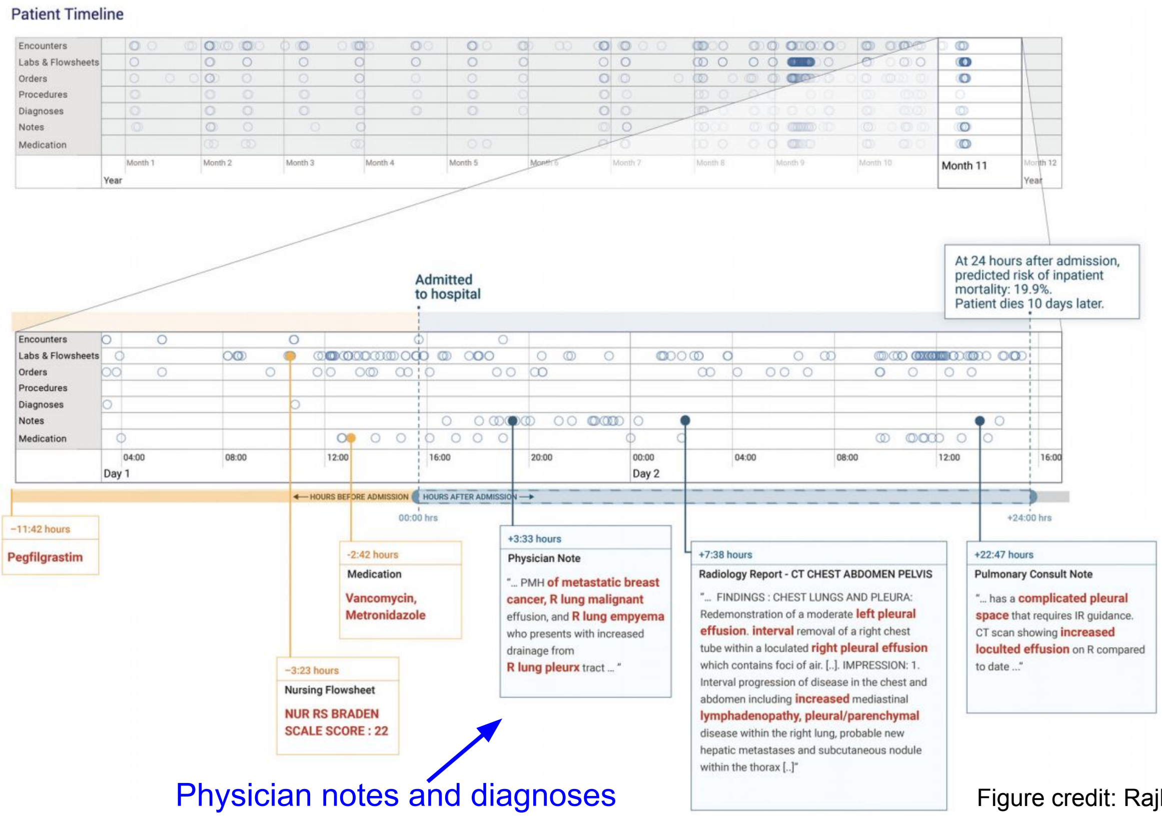
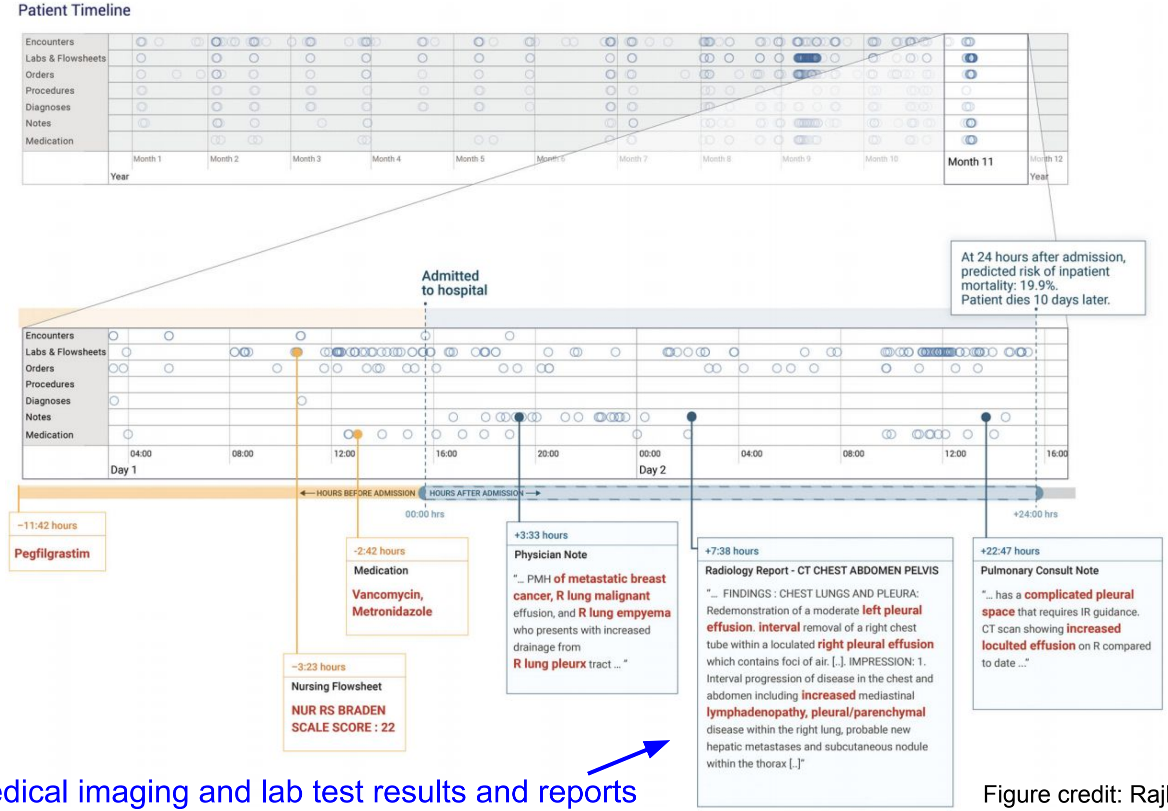


Figure credit: Rajkomar et al. 2018

What are electronic health records?

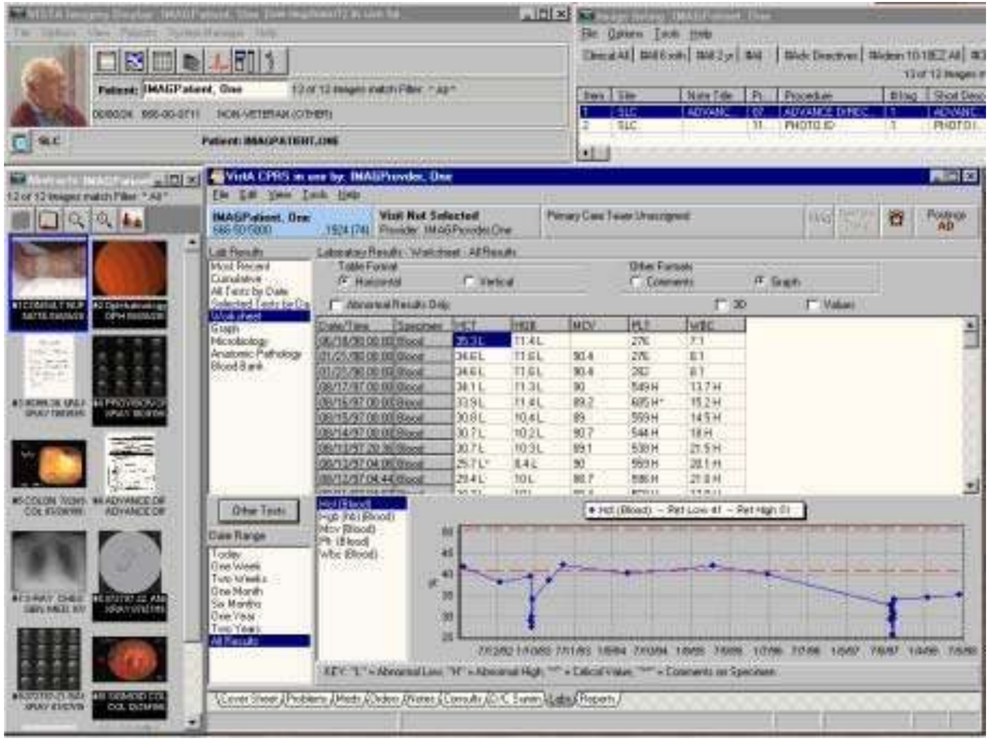
Patient chart in digital form, containing medical and treatment history



Medical imaging and lab test results and reports

Figure credit: Rajkomar et al. 2018

What are electronic health records?



1960s: invention
1980s: increased effort
2009: **HITECH Act** (Health Information Technology for Economic and Clinical Health Act) -- financial incentives for health care providers to adopt EHR

EHR adoption in the US (hospitals)

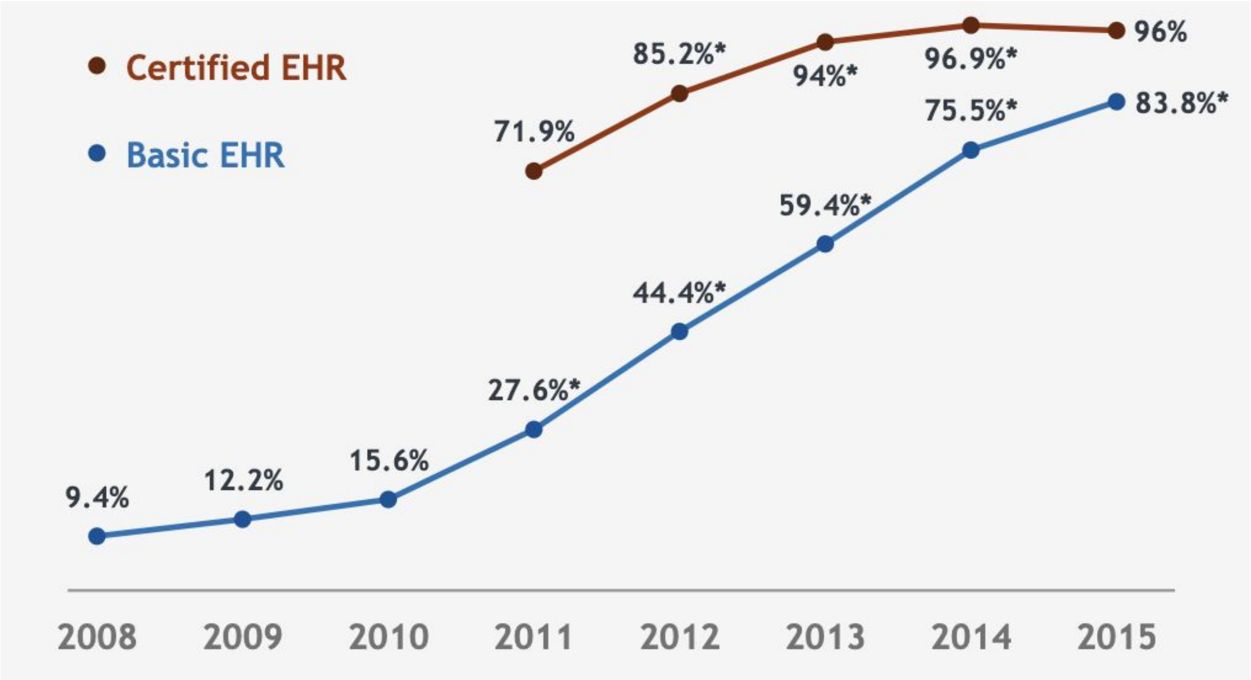


Figure credit: https://dashboard.healthit.gov/evaluations/images/db-35-figure1.svg

EHR adoption in the US (hospitals)

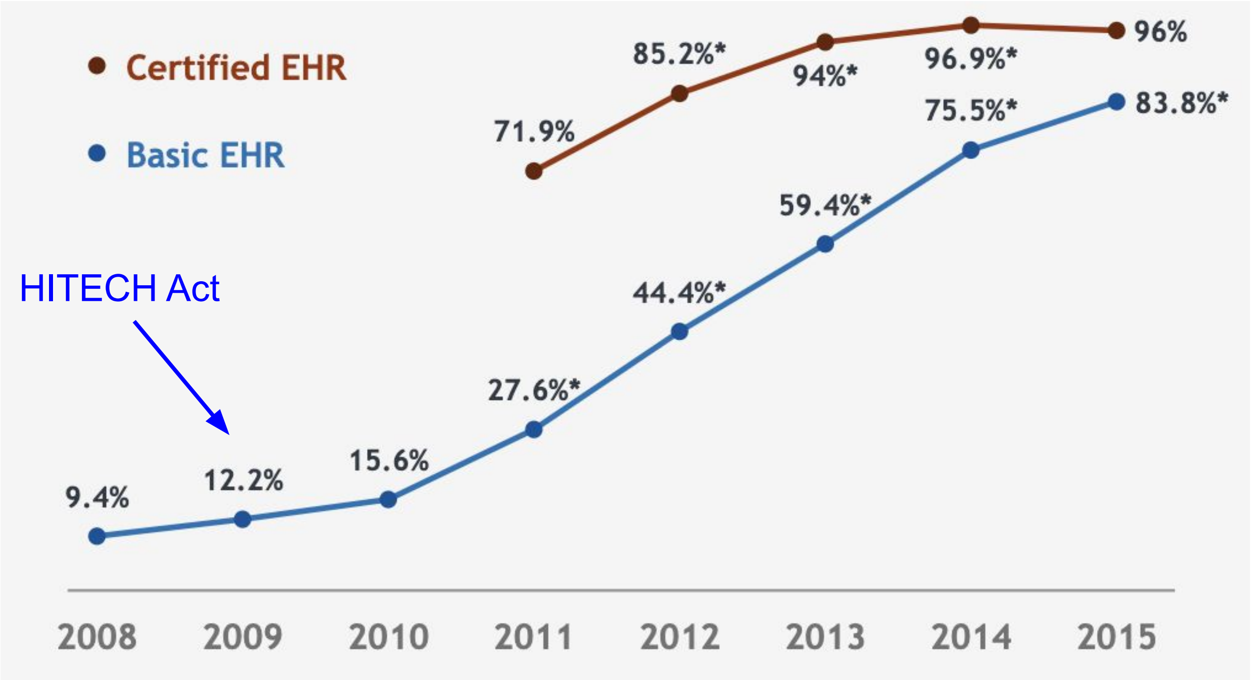


Figure credit: https://dashboard.healthit.gov/evaluations/images/db-35-figure1.svg

EHR adoption in the US (office-based physicians)

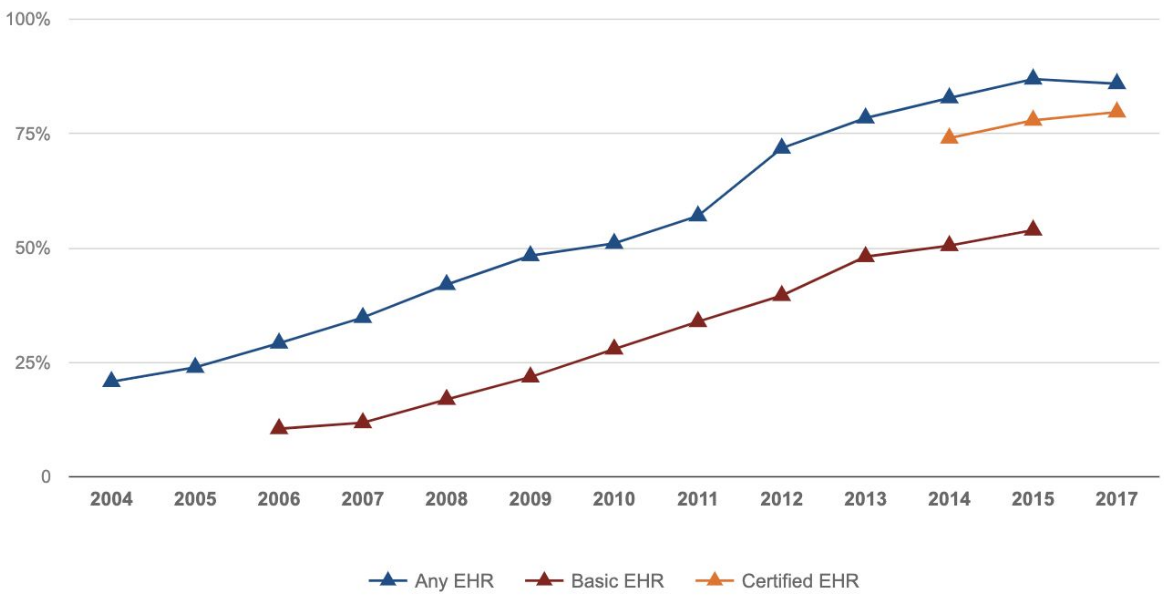
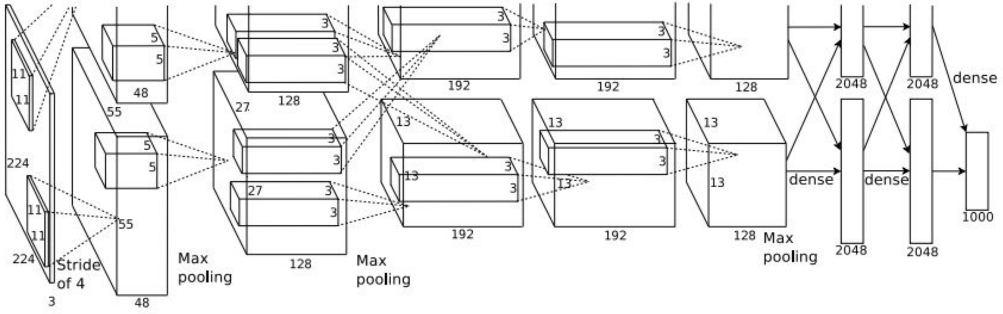


Figure credit: https://dashboard.healthit.gov/quickstats/pages/physician-ehr-adoption-trends.php

Convergence of key ingredients of deep learning

Algorithms

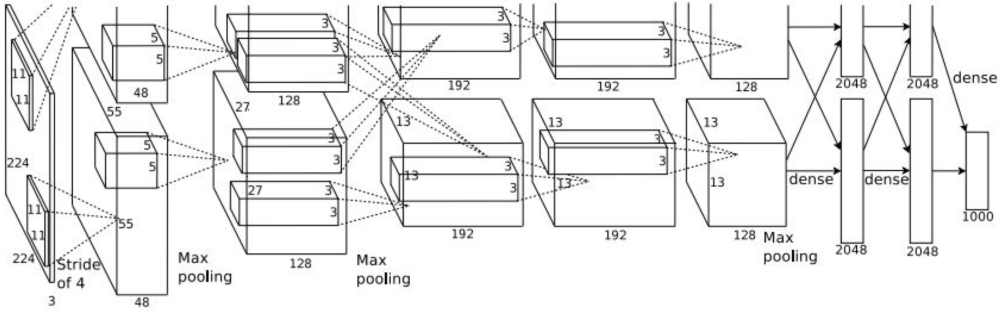


Compute

Data

Convergence of key ingredients of deep learning

Algorithms



Compute

Data

A real example of EHR data: MIMIC-III dataset

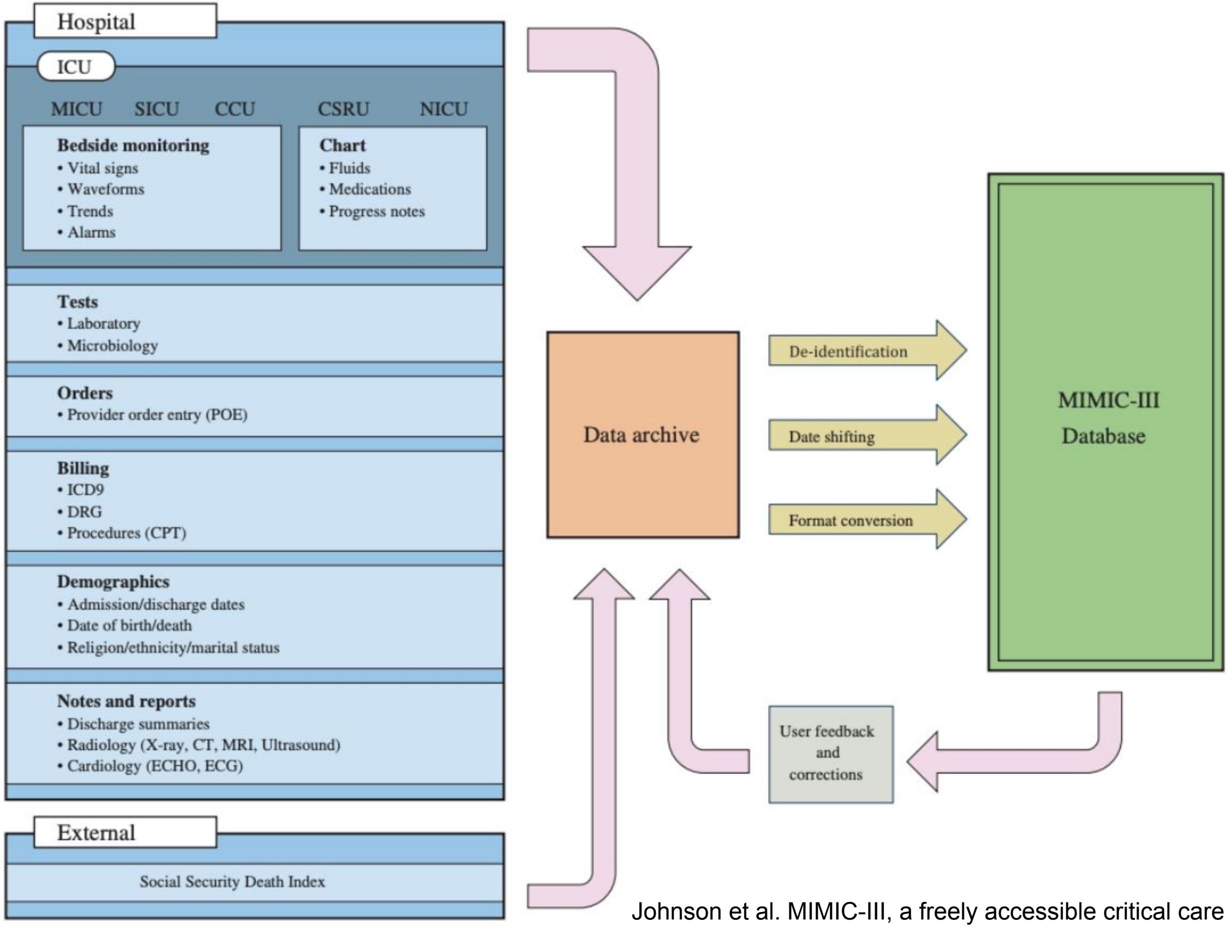
- Open source database of de-identified data for 38,597 adult patients, corresponding to 49,785 hospital admissions
- All patients admitted to critical care units at Beth Israel Deaconess Medical Center (Boston, MA) between 2001 - 2012
- Also 7870 neonates admitted between 2001-2008
- Median hospital stay length: 6.9 days
- Median ICU stay length: 2.1 days
- In-hospital mortality: 11.5%
- Mean of 4579 charted observations and 380 laboratory measurements for each admission

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

A real example of EHR data: MIMIC-III dataset

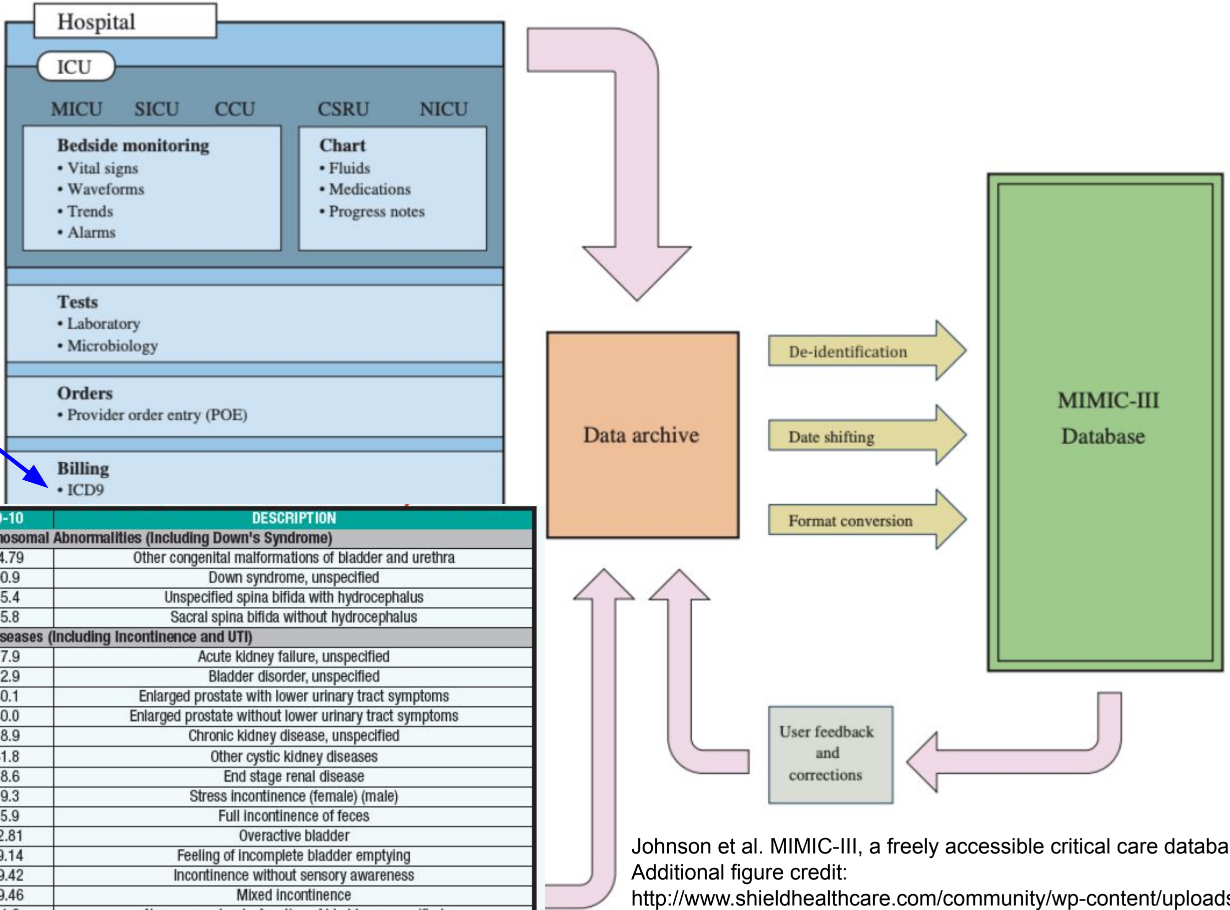
Critical care unit	CCU	CSRU	MICU	SICU	TSICU	Total
Distinct patients, no. (% of total admissions)	5,674 (14.7%)	8,091 (20.9%)	13,649 (35.4%)	6,372 (16.5%)	4,811 (12.5%)	38,597 (100%)
Hospital admissions, no. (% of total admissions)	7,258 (14.6%)	9,156 (18.4%)	19,770 (39.7%)	8,110 (16.3%)	5,491 (11.0%)	49,785 (100%)
Distinct ICU stays, no. (% of total admissions)	7,726 (14.5%)	9,854 (18.4%)	21,087 (39.5%)	8,891 (16.6%)	5,865 (11.0%)	53,423 (100%)
Age, years, median (Q1-Q3)	70.1 (58.4–80.5)	67.6 (57.6–76.7)	64.9 (51.7–78.2)	63.6 (51.4–76.5)	59.9 (42.9–75.7)	65.8 (52.8–77.8)
Gender, male, % of unit stays	4,203 (57.9%)	6,000 (65.5%)	10,193 (51.6%)	4,251 (52.4%)	3,336 (60.7%)	27,983 (55.9%)
ICU length of stay, median days (Q1-Q3)	2.2 (1.2–4.1)	2.2 (1.2–4.0)	2.1 (1.2–4.1)	2.3 (1.3–4.9)	2.1 (1.2–4.6)	2.1 (1.2–4.6)
Hospital length of stay, median days (Q1-Q3)	5.8 (3.1–10.0)	7.4 (5.2–11.4)	6.4 (3.7–11.7)	7.9 (4.4–14.2)	7.4 (4.1–13.6)	6.9 (4.1–11.9)
ICU mortality, percent of unit stays	685 (8.9%)	353 (3.6%)	2,222 (10.5%)	813 (9.1%)	492 (8.4%)	4,565 (8.5%)
Hospital mortality, percent of unit stays	817 (11.3%)	424 (4.6%)	2,859 (14.5%)	1,020 (12.6%)	628 (11.4%)	5,748 (11.5%)

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



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

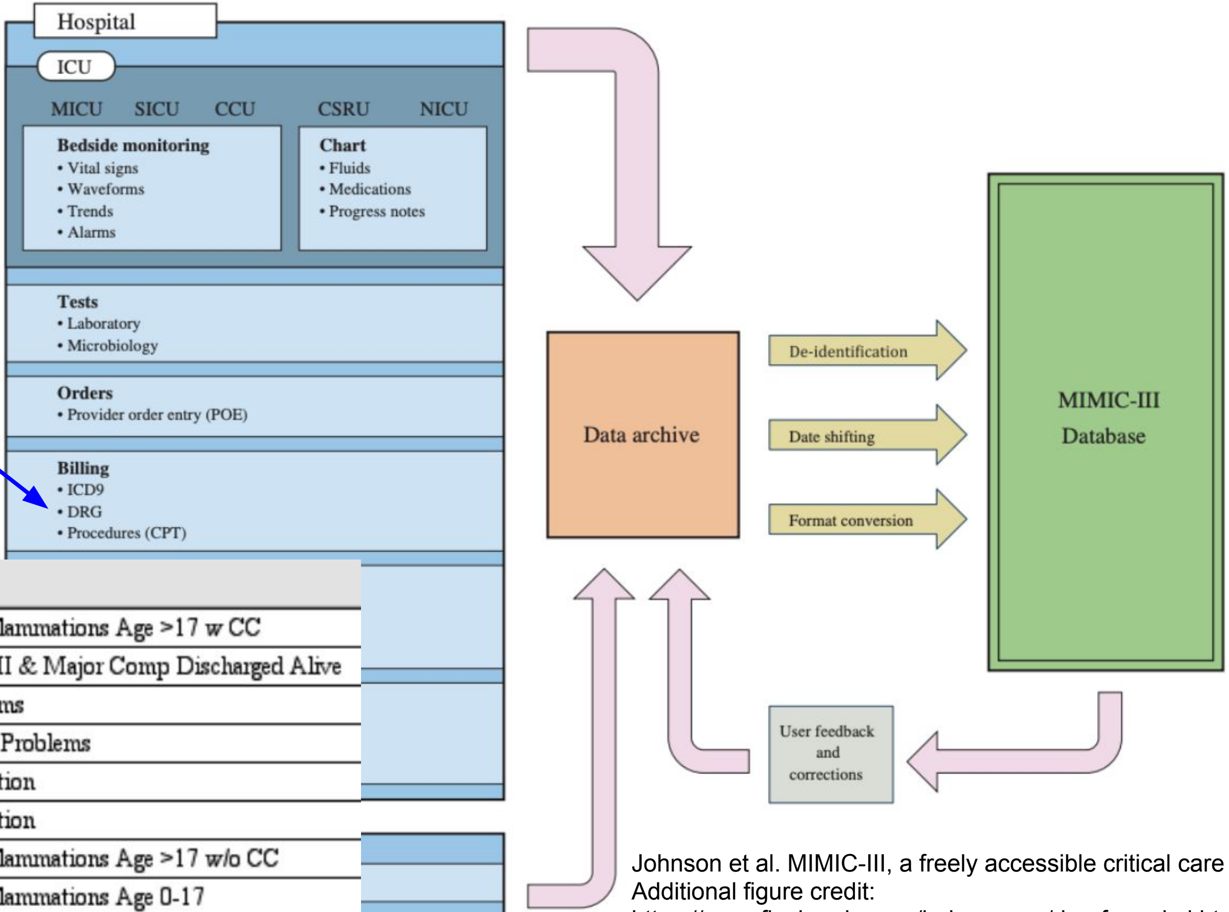
ICD9 (International classification of diseases): Diagnosis codes



ICD-9	DESCRIPTION	ICD-10	DESCRIPTION
Congenital Malformations and Chromosomal Abnormalities (Including Down's Syndrome)			
753.8	Cystourethral anom NEC	Q64.79	Other congenital malformations of bladder and urethra
758.0	Down's syndrome	Q90.9	Down syndrome, unspecified
741.00	Spin bif w hydroceph NOS	Q05.4	Unspecified spina bifida with hydrocephalus
741.90	Spina bifida	Q05.8	Sacral spina bifida without hydrocephalus
Genitourinary System Diseases (Including Incontinence and UTI)			
584.9	Acute kidney failure NOS	N17.9	Acute kidney failure, unspecified
596.9	Bladder disorder NOS	N32.9	Bladder disorder, unspecified
600.01	BPH w urinary obs/LUTS	N40.1	Enlarged prostate with lower urinary tract symptoms
600.00	BPH w/o urinary obs/LUTS	N40.0	Enlarged prostate without lower urinary tract symptoms
585.9	Chronic kidney dis NOS	N18.9	Chronic kidney disease, unspecified
753.19	Cystic kidney diseas NEC	Q61.8	Other cystic kidney diseases
585.6	End stage renal disease	N18.6	End stage renal disease
625.6	Fem stress incontinence	N39.3	Stress incontinence (female) (male)
787.60	Full incontinence-feces	R15.9	Full incontinence of feces
596.51	Hypertonicity of bladder	N32.81	Overactive bladder
788.21	Incmplet bldder emptying	R39.14	Feeling of incomplete bladder emptying
788.34	Incontnce wo sensr aware	N39.42	Incontinence without sensory awareness
788.33	Mixed incontinence	N39.46	Mixed incontinence
596.54	Neurogenic bladder NOS	N31.9	Neuromuscular dysfunction of bladder, unspecified
788.39	Oth urinry incontinence	N39.498	Other specified urinary incontinence

Johnson et al. MIMIC-III, a freely accessible critical care database. 2016.
Additional figure credit:
http://www.shieldhealthcare.com/community/wp-content/uploads/2015/08/ICD-9-to-ICD-10-Conversion-Guide-Page-1.jpg

DRG (diagnosis related group): Higher-level codes describing patient groups w/ similar hospital resource use

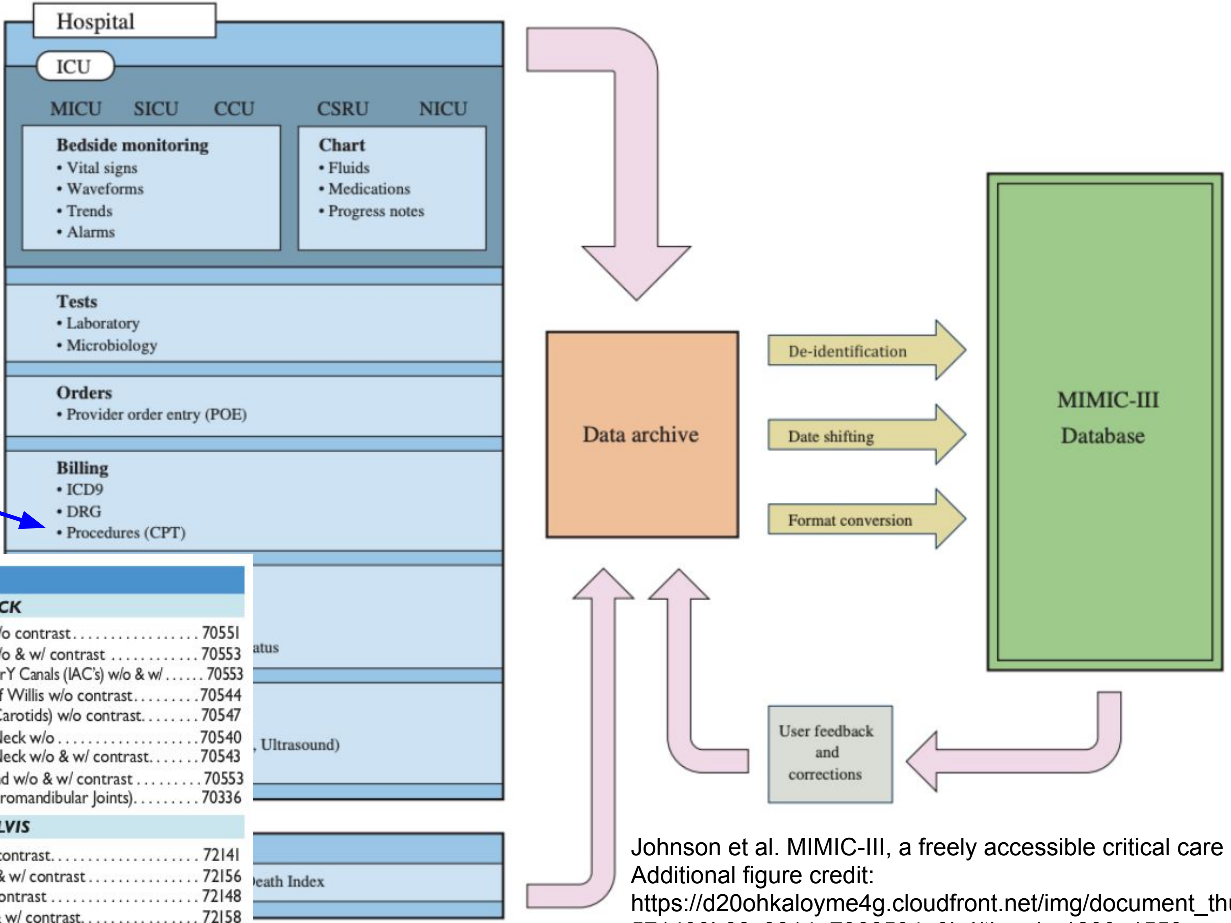


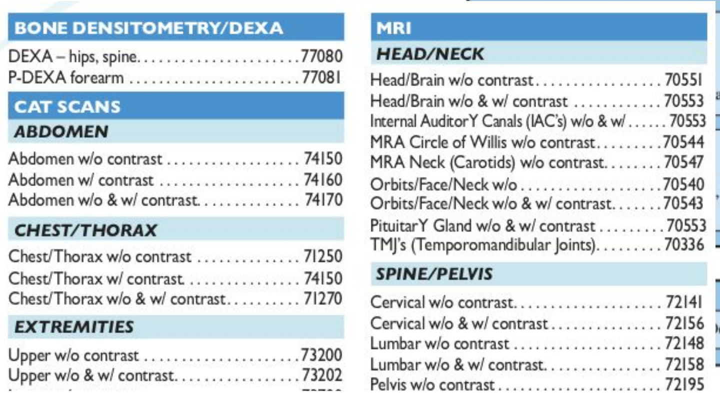
DRG Code and Description

079	Respiratory Infections & Inflammations Age >17 w CC
121	Circulatory Disorders w AMI & Major Comp Discharged Alive
387	Prematurity w Major Problems
389	Full Term Neonate w Major Problems
489	HIV w Major Related Condition
489	HIV w Major Related Condition
080	Respiratory Infections & Inflammations Age >17 w/o CC
081	Respiratory Infections & Inflammations Age 0-17

Johnson et al. MIMIC-III, a freely accessible critical care database. 2016.
Additional figure credit:
https://www.flashcode.com/help_pages/drg_from_icd.html

CPT (Current procedural terminology): procedures and services codes



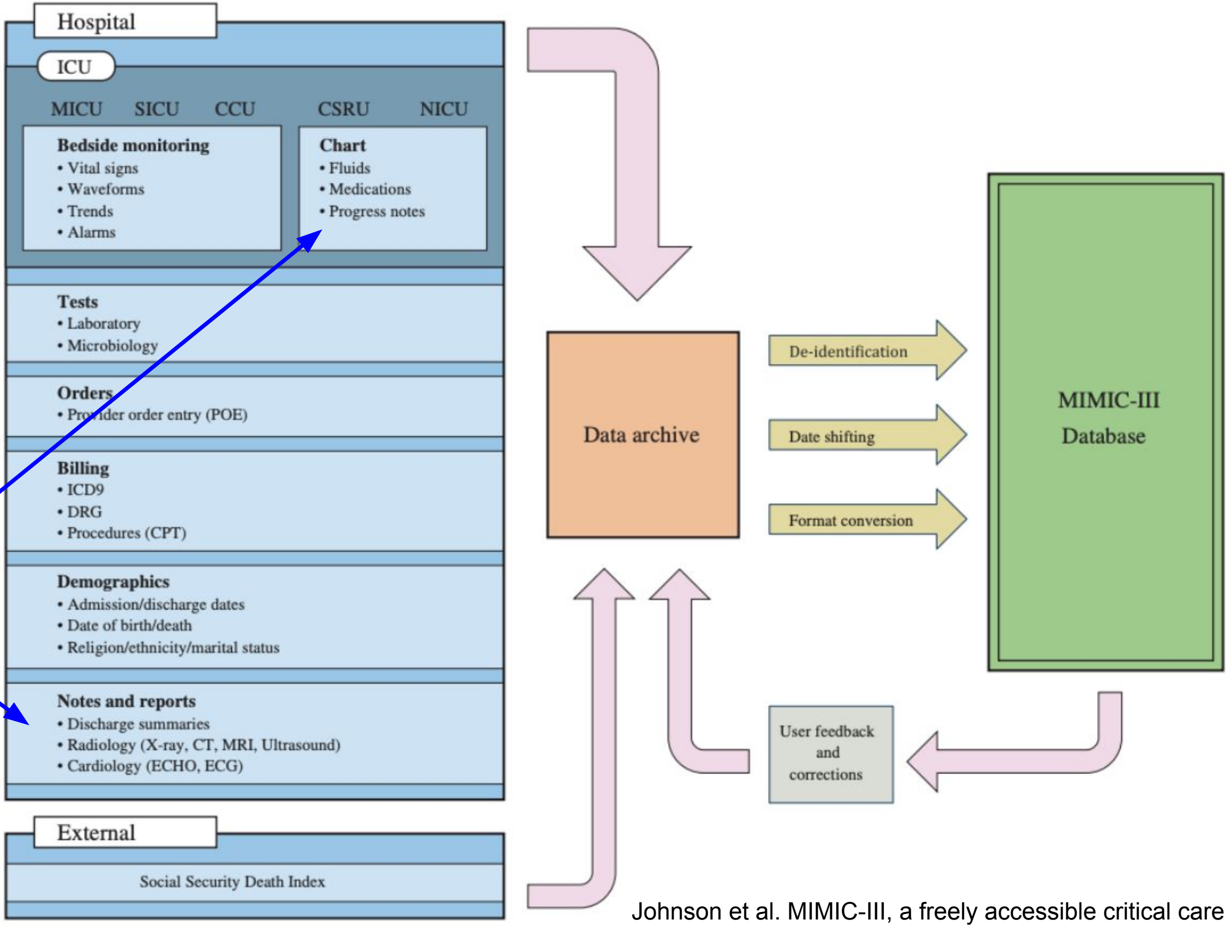


BONE DENSITOMETRY/DEXA	
DEXA – hips, spine	77080
P-DEXA forearm	77081
CAT SCANS	
ABDOMEN	
Abdomen w/o contrast	74150
Abdomen w/ contrast	74160
Abdomen w/o & w/ contrast	74170
CHEST/THORAX	
Chest/Thorax w/o contrast	71250
Chest/Thorax w/ contrast	74150
Chest/Thorax w/o & w/ contrast	71270
EXTREMITIES	
Upper w/o contrast	73200
Upper w/o & w/ contrast	73202

MRI	
HEAD/NECK	
Head/Brain w/o contrast	70551
Head/Brain w/o & w/ contrast	70553
Internal AuditorY Canals (IAC's) w/o & w/	70553
MRA Circle of Willis w/o contrast	70544
MRA Neck (Carotids) w/o contrast	70547
Orbits/Face/Neck w/o	70540
Orbits/Face/Neck w/o & w/ contrast	70543
PituitarY Gland w/o & w/ contrast	70553
TMJ's (Temporomandibular Joints)	70336
SPINE/PELVIS	
Cervical w/o contrast	72141
Cervical w/o & w/ contrast	72156
Lumbar w/o contrast	72148
Lumbar w/o & w/ contrast	72158
Pelvis w/o contrast	72195

Johnson et al. MIMIC-III, a freely accessible critical care database. 2016.
Additional figure credit:
https://d20ohkaloyme4g.cloudfront.net/img/document_thumbnails/e570ad571499b88c8814e7366594e9bd/thumb_1200_1553.png

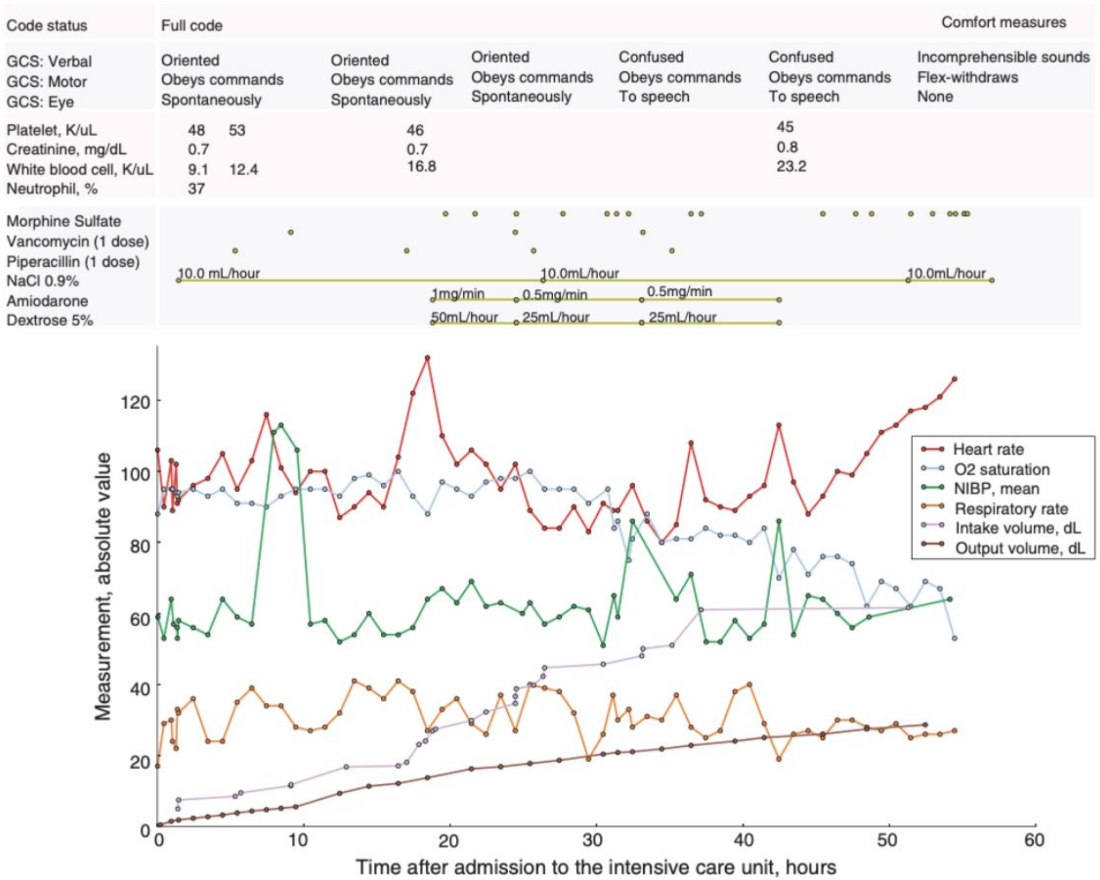
Free text data



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

Critical care unit	CCU stays, No. (% by unit)	CSRU stays, No. (% by unit)	MICU stays, No. (% by unit)	SICU stays, No. (% by unit)	TSICU stays, No. (% by unit)	Total stays, No. (% by unit)
Infectious and parasitic diseases, i.e., septicemia, other infectious and parasitic diseases, etc., (001–139)	305 (4.2%)	72 (0.8%)	3,229 (16.7%)	448 (5.6%)	152 (2.8%)	4,206 (8.6%)
Neoplasms of digestive organs and intrathoracic organs, etc., (140–239)	126 (1.8%)	287 (3.2%)	1,415 (7.3%)	1,225 (15.3%)	466 (8.6%)	3,519 (7.2%)
Endocrine, nutritional, metabolic, and immunity (240–279)	104 (1.4%)	36 (0.4%)	985 (5.1%)	178 (2.2%)	54 (1.0%)	1,357 (2.8%)
Diseases of the circulatory system, i.e., ischemic heart diseases, diseases of pulmonary circulation, dysrhythmias, heart failure, cerebrovascular diseases, etc., (390–459)	5,131 (71.4%)	7,138 (78.6%)	2,638 (13.6%)	2,356 (29.5%)	684 (12.6%)	17,947 (36.6%)
Pulmonary diseases, i.e., pneumonia and influenza, chronic obstructive pulmonary disease, etc., (460–519)	416 (5.8%)	141 (1.6%)	3,393 (17.5%)	390 (4.9%)	225 (4.1%)	4,565 (9.3%)
Diseases of the digestive system (520–579)	264 (3.7%)	157 (1.7%)	3,046 (15.7%)	1,193 (14.9%)	440 (8.1%)	5,100 (10.4%)
Diseases of the genitourinary system, i.e., nephritis, nephrotic syndrome, nephrosis, and other diseases of the genitourinary system (580–629)	130 (1.8%)	14 (0.2%)	738 (3.8%)	101 (1.3%)	31 (0.6%)	1,014 (2.1%)
Trauma (800–959)	97 (1.3%)	494 (5.4%)	480 (2.5%)	836 (10.5%)	2,809 (51.7%)	4,716 (9.6%)
Poisoning by drugs and biological substances (960–979)	50 (0.7%)	2 (0.0%)	584 (3.0%)	58 (0.7%)	11 (0.2%)	705 (1.4%)
Other	565 (7.9%)	739 (8.1%)	2,883 (14.9%)	1,204 (15.1%)	563 (10.4%)	5,954 (12.1%)
Total	7,188 (14.6%)	9,080 (18.5%)	19,391 (39.5%)	7,989 (16.3%)	5,435 (11.1%)	49,083 (100%)

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



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

Examples of prediction tasks: phenotypes

- What conditions a patient has
- Useful for patient treatment and risk monitoring

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

Examples of prediction tasks: in-hospital mortality

- Whether patient will die in the hospital
- Early detection of at-risk patients can improve outcomes

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

Examples of prediction tasks: decompensation

- Whether patient will die in the next 24 hours
- Also for early detection, related to in-hospital mortality

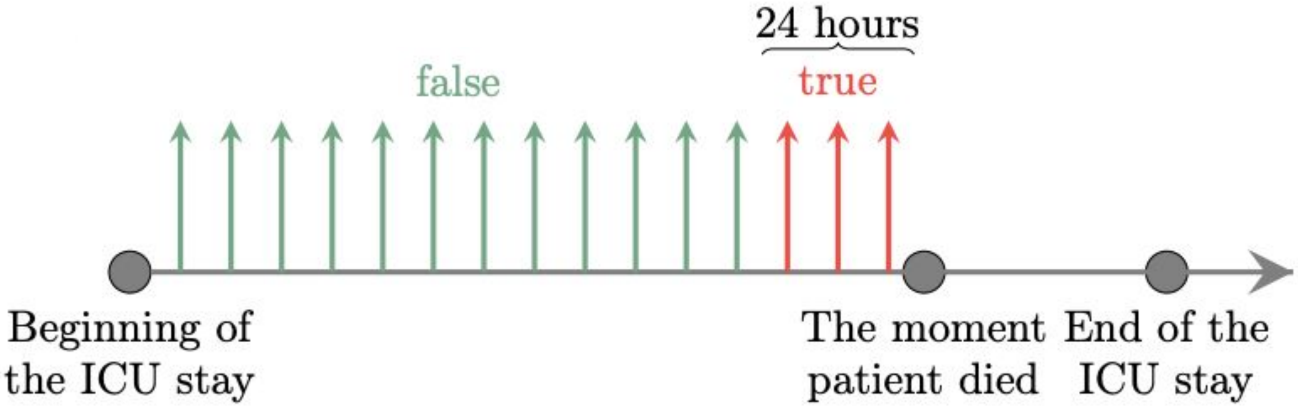


Figure credit: Harutyunyan et al. Multitask learning and benchmarking with clinical time series data. 2019.

Examples of prediction tasks: length-of-stay

- How much longer the patient is expected to stay in the ICU
- Useful for measuring patient acuity and resource management

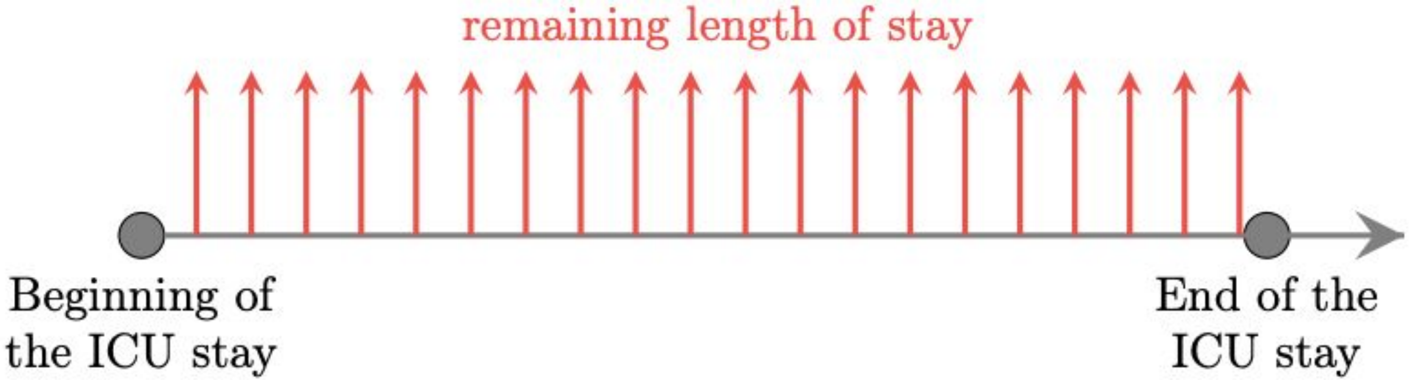


Figure credit: Harutyunyan et al. Multitask learning and benchmarking with clinical time series data. 2019.

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]

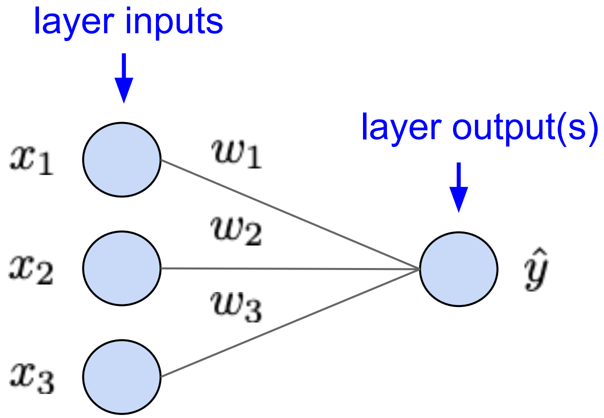
$\hat{y} =$ length-of-stay (days)

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

Our first architecture: a single-layer, fully connected neural network

all inputs of a layer are connected to all outputs of a layer

For simplicity, use a 3-dimensional input (N = 3)



Output: $\hat{y} = w_1 x_1 + w_2 x_2 + w_3 x_3 + b$

$= w^T x + b$

bias term (allows constant shift)

Neural network parameters:

$$W = \{[w_1, w_2, w_3], b\}$$

layer “weights”

layer bias

Often refer to all parameters together as just “weights”. Bias is implicitly assumed.

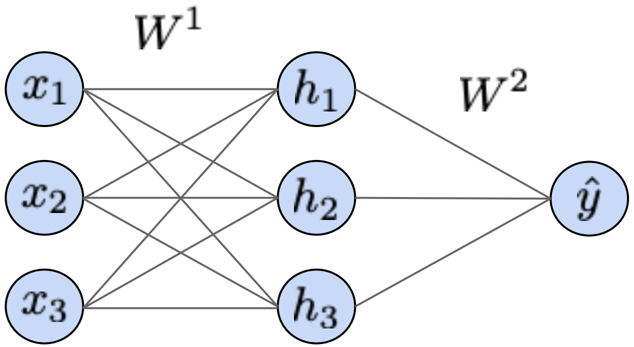
Caveats of our first (simple) neural network architecture:

- Single layer still “shallow”, not yet a “deep” neural network. Will see soon how to stack multiple layers.
- Also equivalent to a linear regression model! But useful base case for deep learning.

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

$$\sigma(a) = \frac{1}{1+e^{-a}}$$

Two-layer fully-connected neural network



$$W^1 = \begin{bmatrix} w^1_{11} & w^1_{12} & w^1_{13} \\ w^1_{21} & w^1_{22} & w^1_{23} \\ w^1_{31} & w^1_{32} & w^1_{33} \end{bmatrix} \quad b^1 = \begin{bmatrix} b^1_1 \\ b^1_2 \\ b^1_3 \end{bmatrix}$$

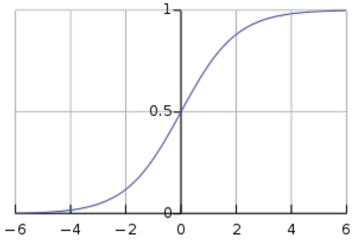
$$W^2 = \begin{bmatrix} w^2_{11} & w^2_{12} & w^2_{13} \end{bmatrix} \quad b^2 = \begin{bmatrix} b^2_1 \end{bmatrix}$$

Output: $h = \sigma(W^1 x + b^1)$

$$\hat{y} = W^2 h + b^2$$

Full function expression:

$$\hat{y} = W^2(\sigma(W^1 x + b^1)) + b^2$$

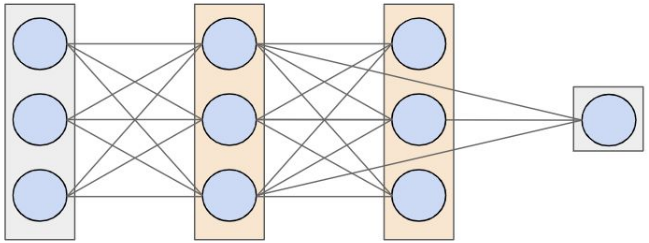


Sigmoid “activation function”

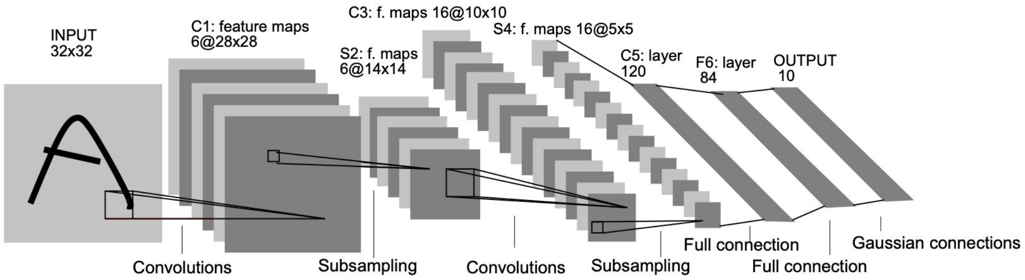
Activation functions introduce non-linearity into the model -- allowing it to represent highly complex functions.

A **fully-connected neural network** (also known as multi-layer perceptron) is a stack of [affine transformation + activation function] layers. There is no activation function at the last layer.

Different classes of neural networks

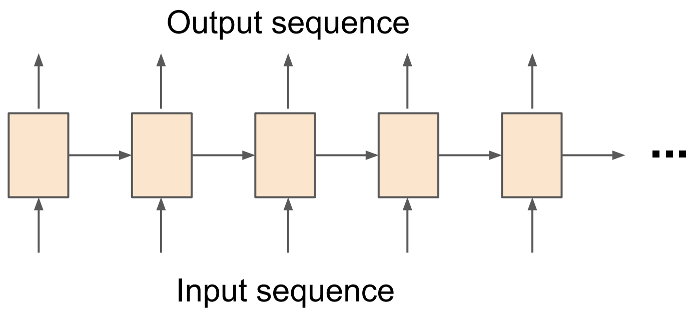


Fully connected neural networks
(linear layers, good for “feature vector” inputs)

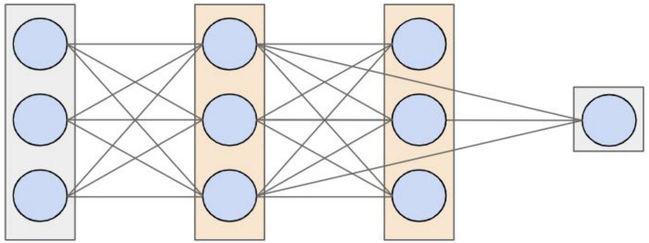


Convolutional neural networks
(convolutional layers, good for image inputs)

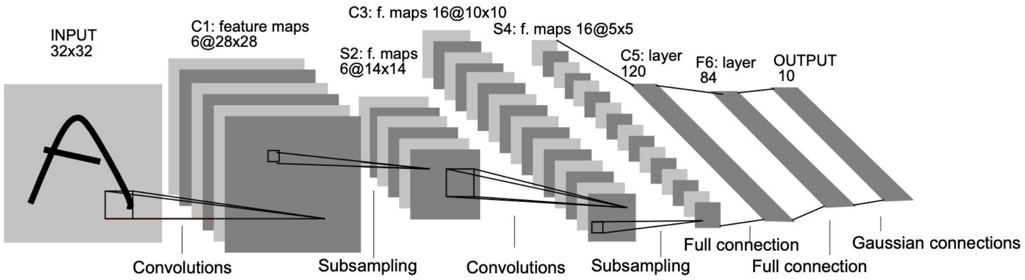
Recurrent neural networks
(linear layers modeling recurrence relation across sequence, good for sequence inputs)



Different classes of neural networks



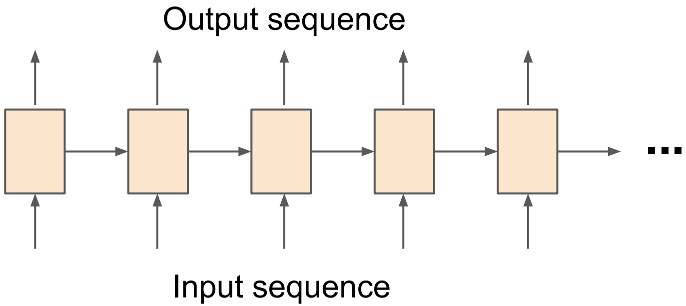
Fully connected neural networks
(linear layers, good for "feature vector" inputs)



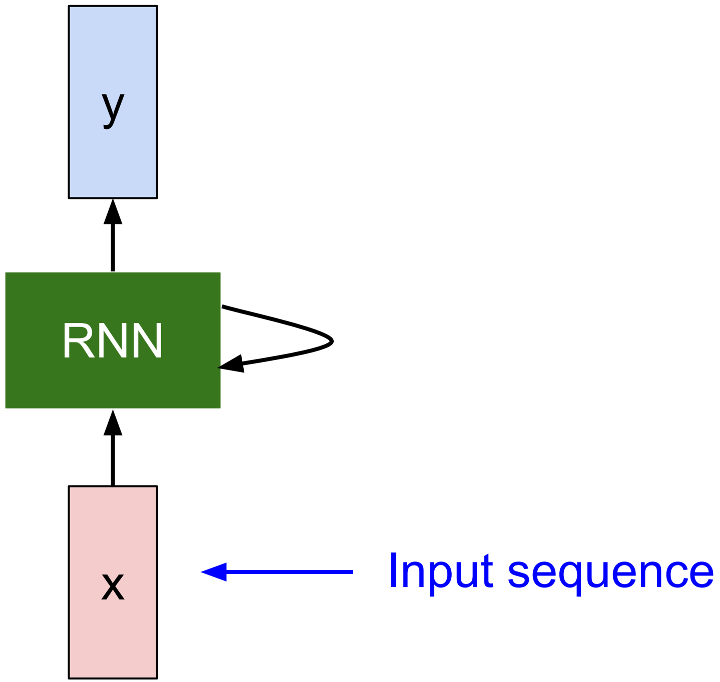
Convolutional neural networks
(convolutional layers, good for image inputs)

Natural fit for clinical time-series data

Recurrent neural networks
(linear layers modeling recurrence relation across sequence, good for sequence inputs)

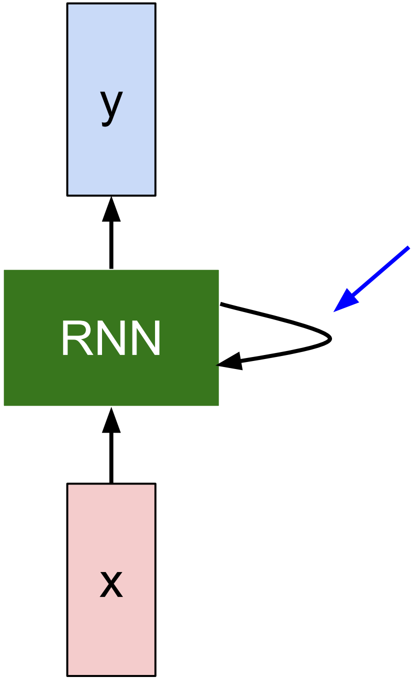


Recurrent Neural Network



Slide credit: CS231n

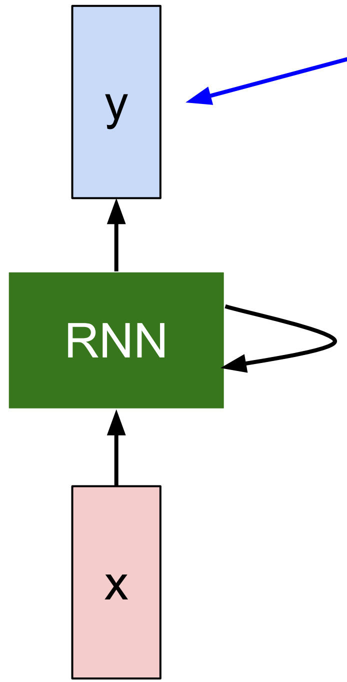
Recurrent Neural Network



Key idea: RNNs have an "internal state" that is updated as a sequence is processed

Slide credit: CS231n

Recurrent Neural Network



Output
(do not necessarily need to have an output at every timestep of the sequence)

Slide credit: CS231n

Recurrent Neural Network

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

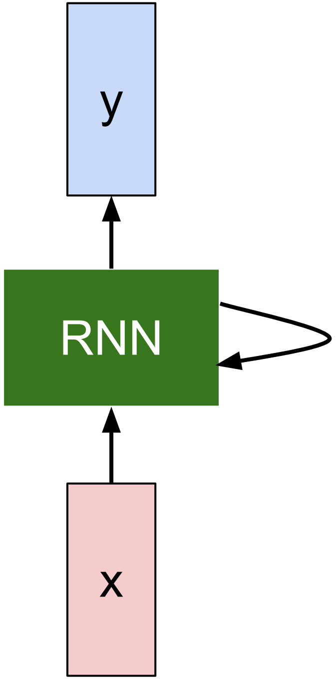
$$\boxed{h_t} = \boxed{f_W}(\boxed{h_{t-1}}, \boxed{x_t})$$

new state

old state

input vector at some time step

some function with parameters W



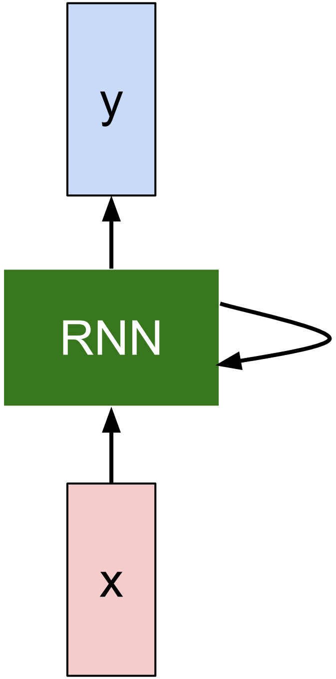
Slide credit: CS231n

Recurrent Neural Network

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

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

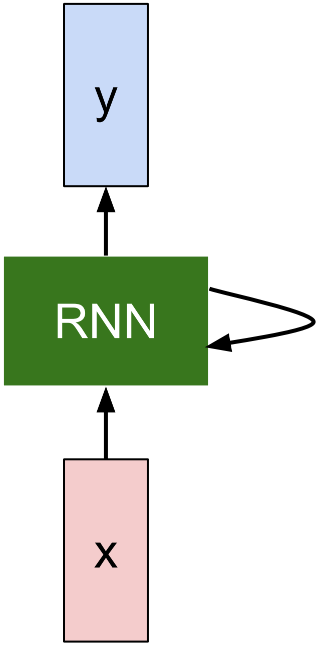
Notice: the same function and the same set of parameters are used at every time step.



Slide credit: CS231n

(Vanilla) Recurrent Neural Network

The state consists of a single *"hidden"* vector **h**:



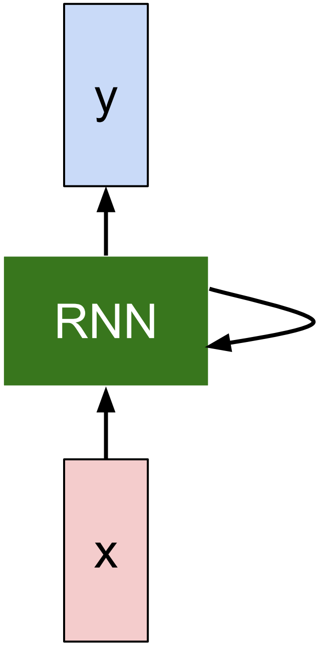
$$h_t = f_W(h_{t-1}, x_t)$$

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

Slide credit: CS231n

(Vanilla) Recurrent Neural Network

The state consists of a single *"hidden"* vector **h**:



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

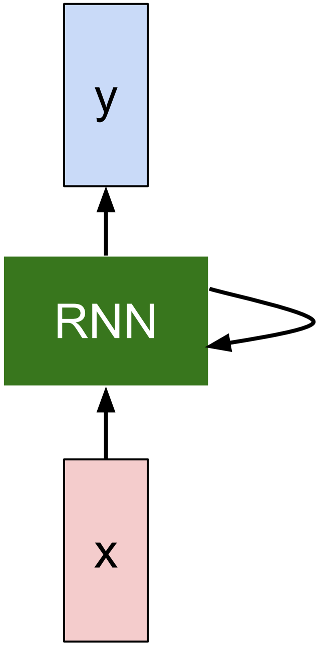
$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$

$$y_t = W_{hy}h_t$$

Slide credit: CS231n

(Vanilla) Recurrent Neural Network

The state consists of a single *"hidden"* vector **h**:



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

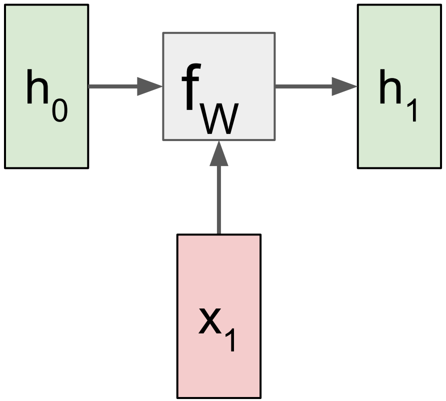
$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$

$$y_t = W_{hy}h_t$$

Fully connected layers

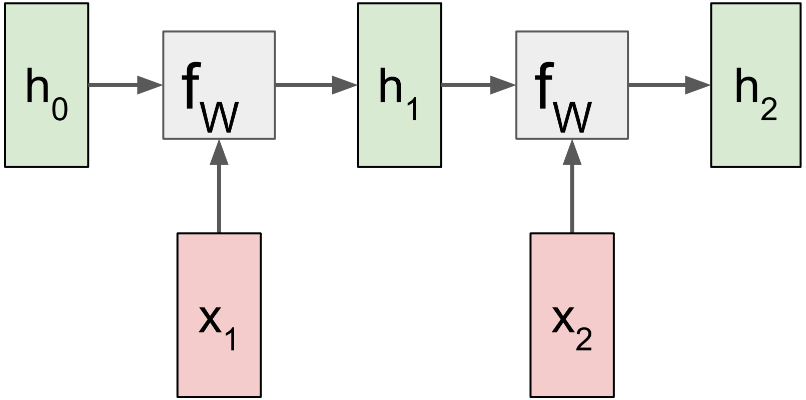
Slide credit: CS231n

RNN: Computational Graph



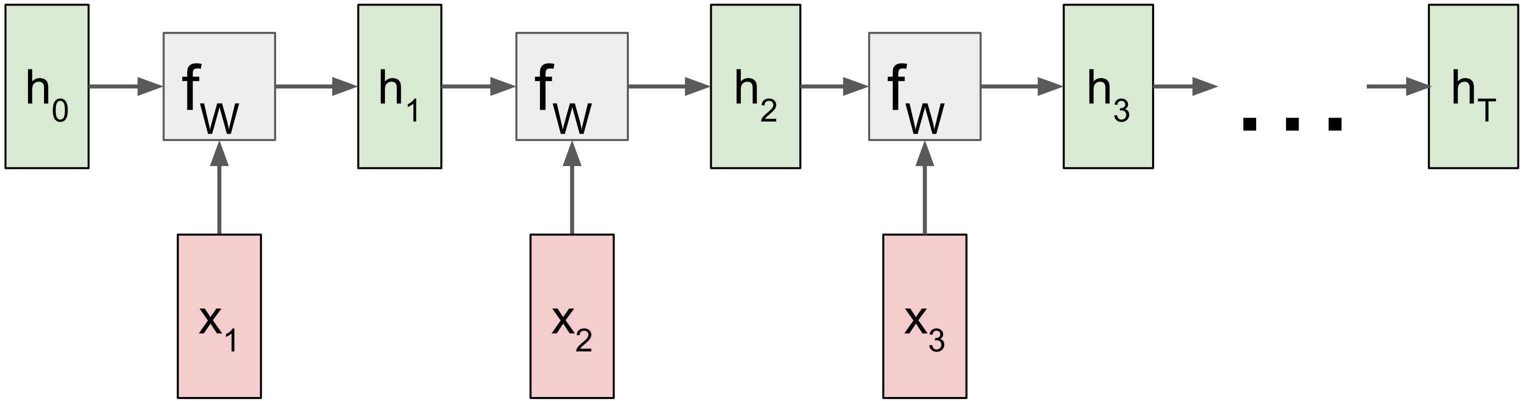
Slide credit: CS231n

RNN: Computational Graph



Slide credit: CS231n

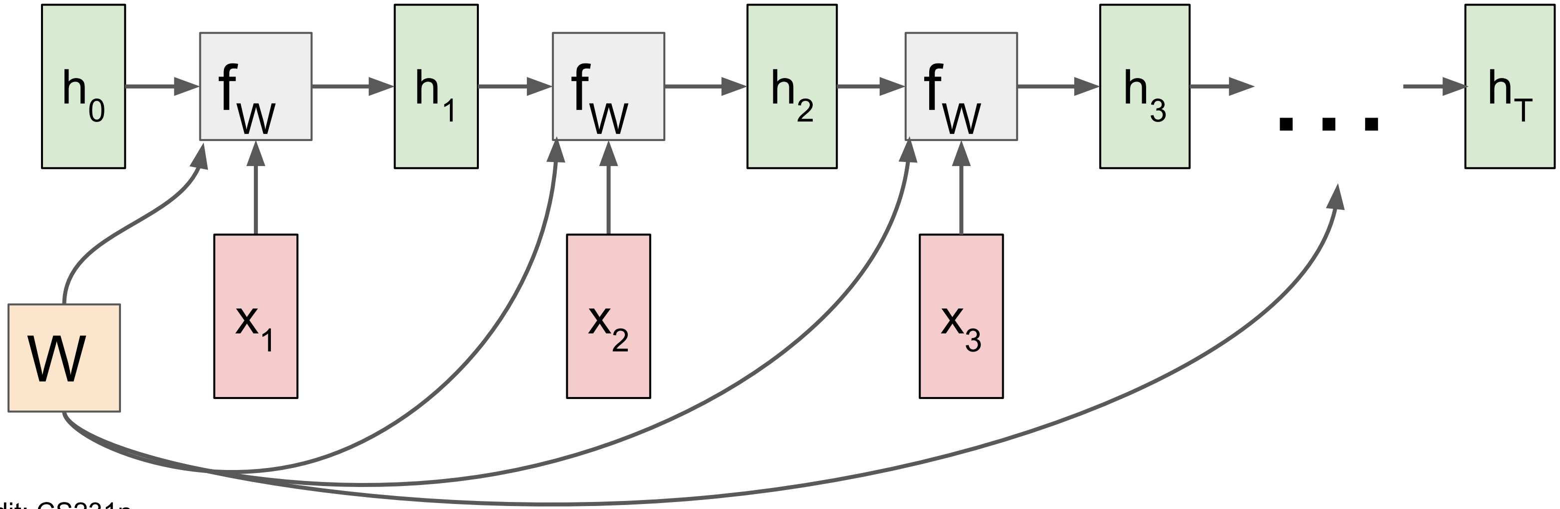
RNN: Computational Graph



Slide credit: CS231n

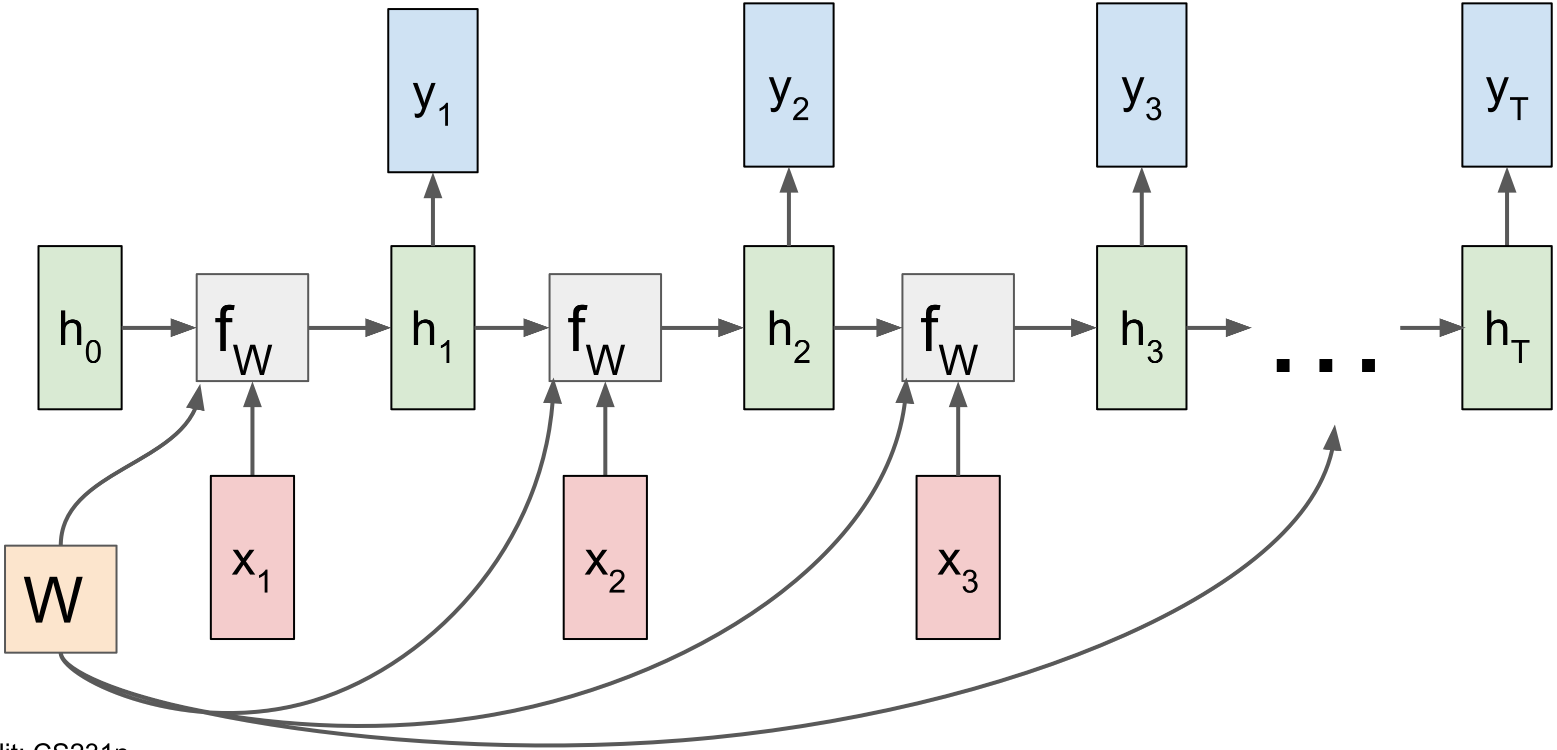
RNN: Computational Graph

Re-use the same weight matrix at every time-step



Slide credit: CS231n

RNN: Computational Graph: Many to Many

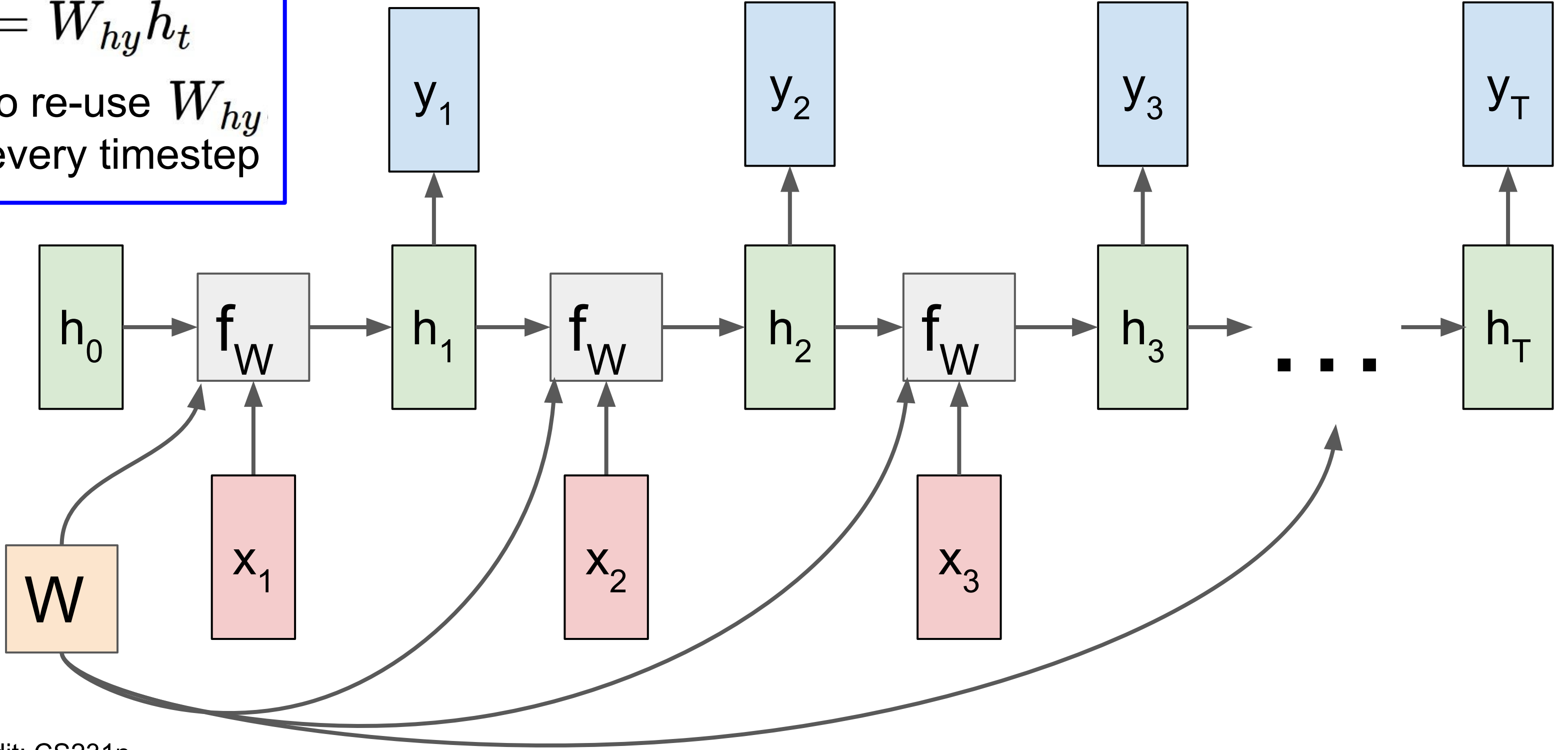


Slide credit: CS231n

RNN: Computational Graph: Many to Many

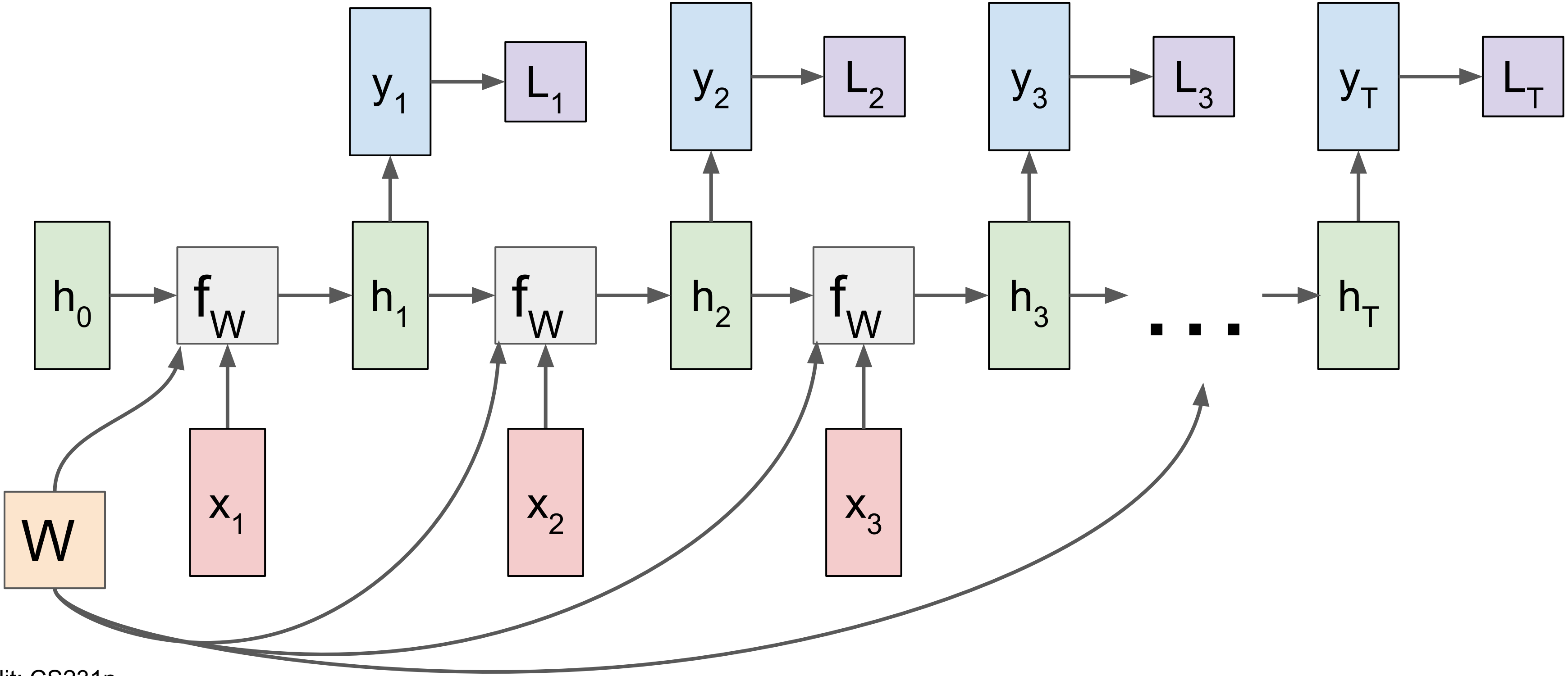
$y_t = W_{hy}h_t$

Also re-use W_{hy} at every timestep



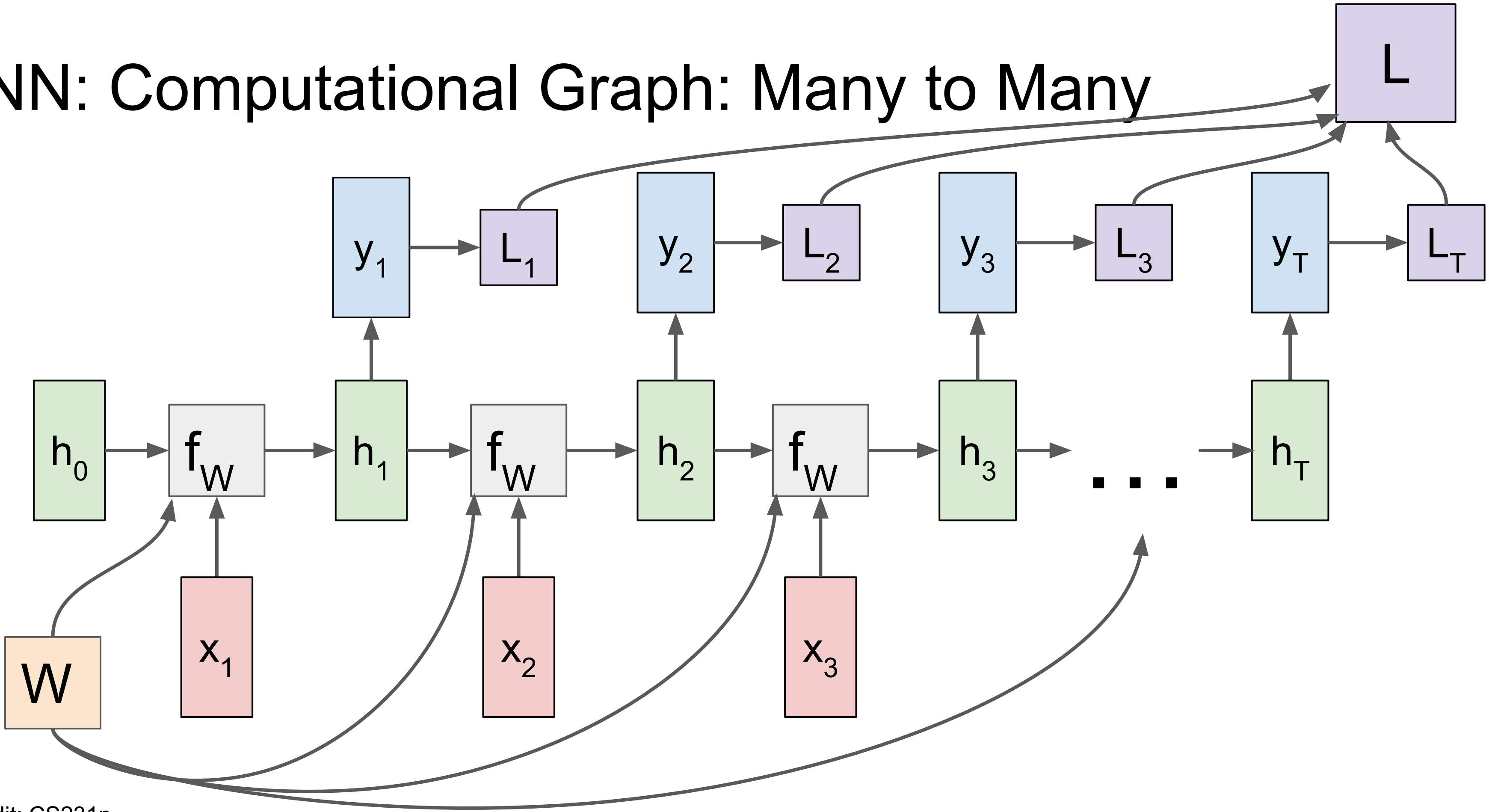
Slide credit: CS231n

RNN: Computational Graph: Many to Many



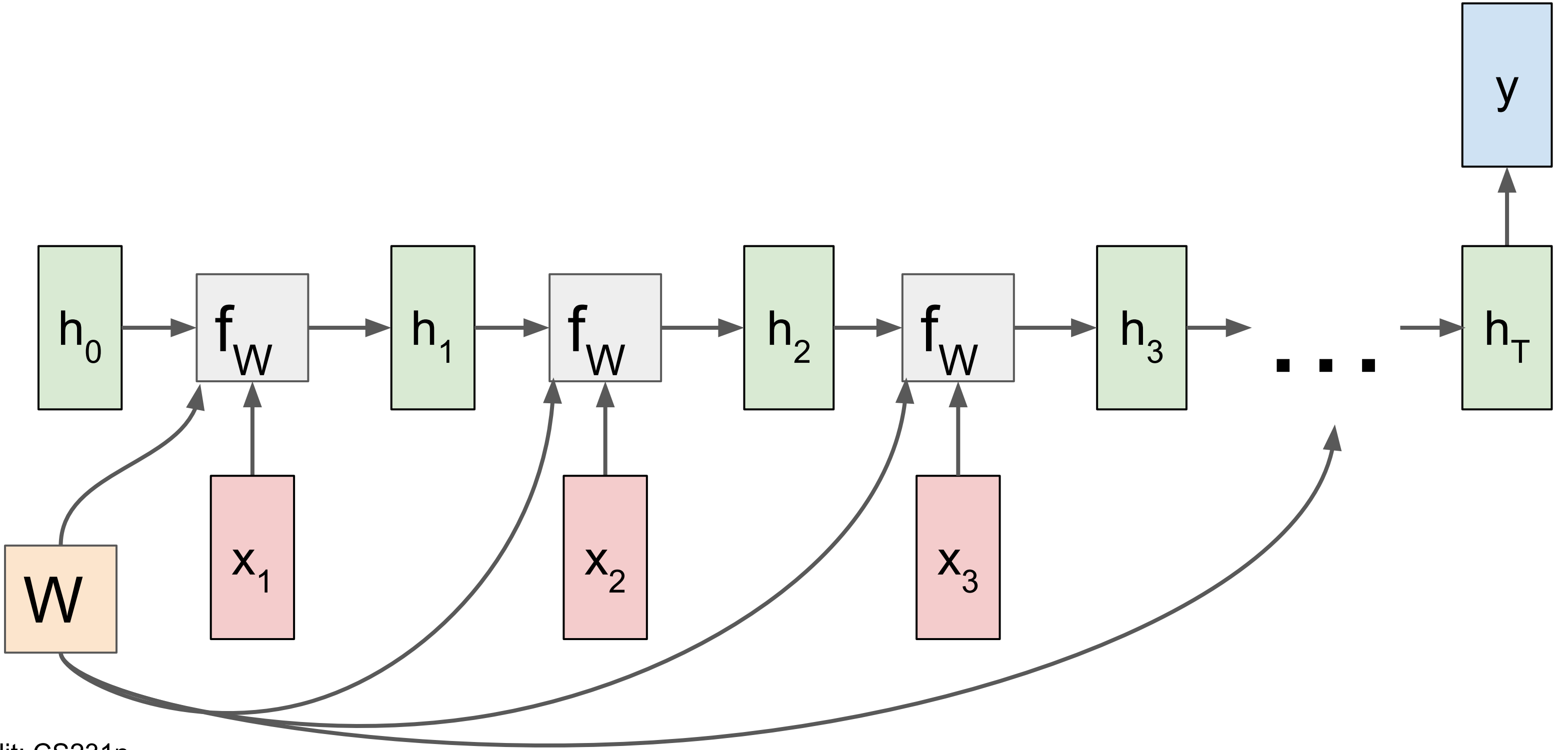
Slide credit: CS231n

RNN: Computational Graph: Many to Many



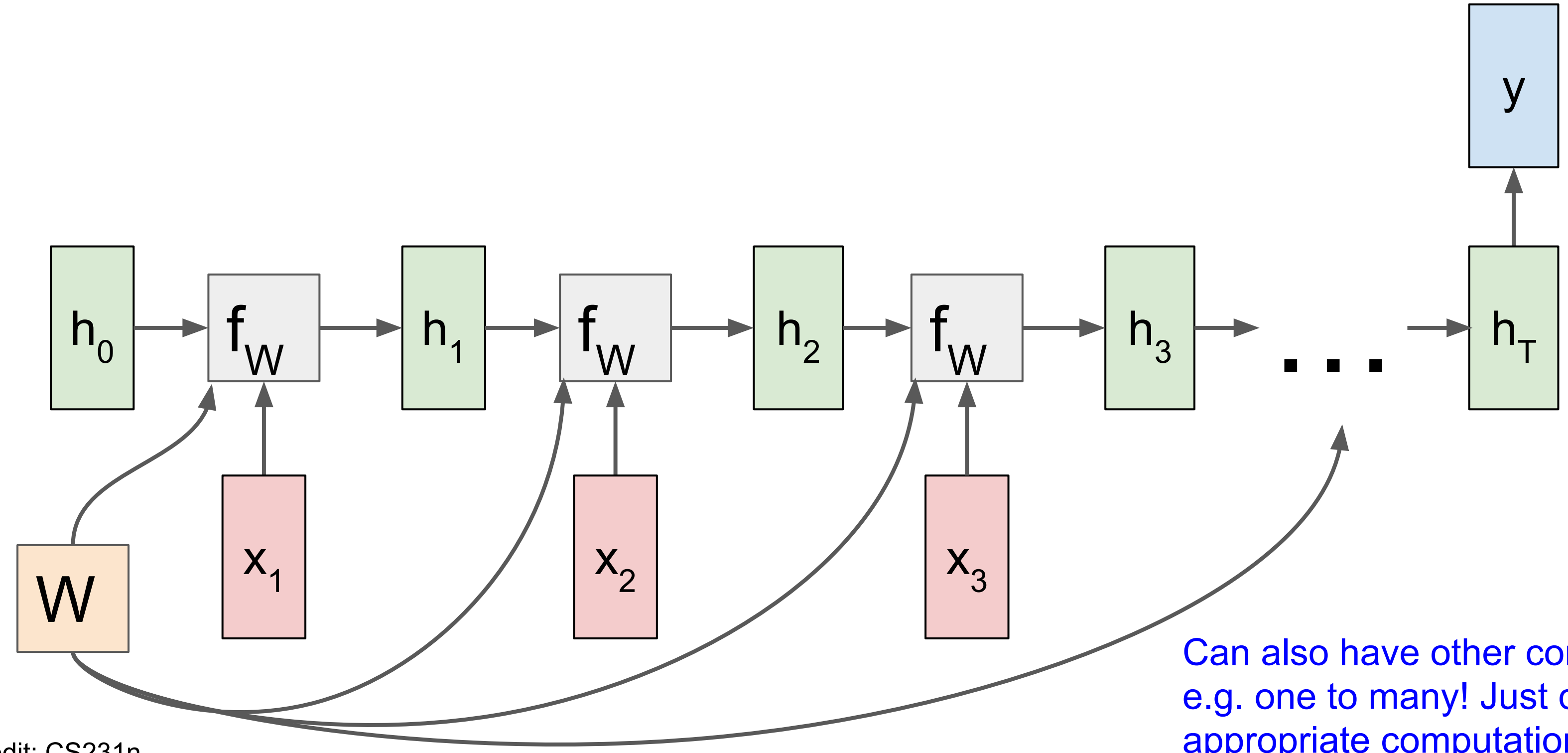
Slide credit: CS231n

RNN: Computational Graph: Many to One



Slide credit: CS231n

RNN: Computational Graph: Many to One

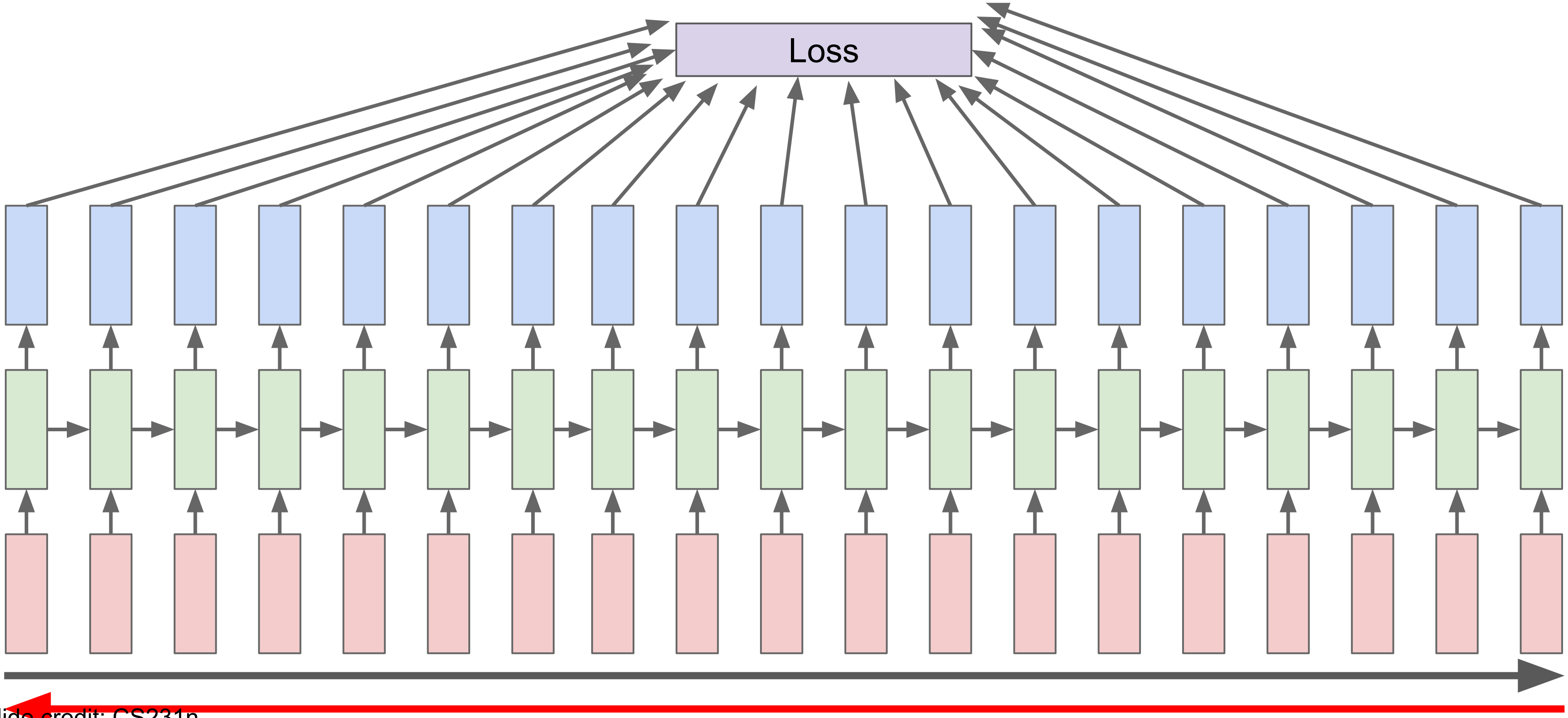


Can also have other configurations e.g. one to many! Just define appropriate computational graph.

Slide credit: CS231n

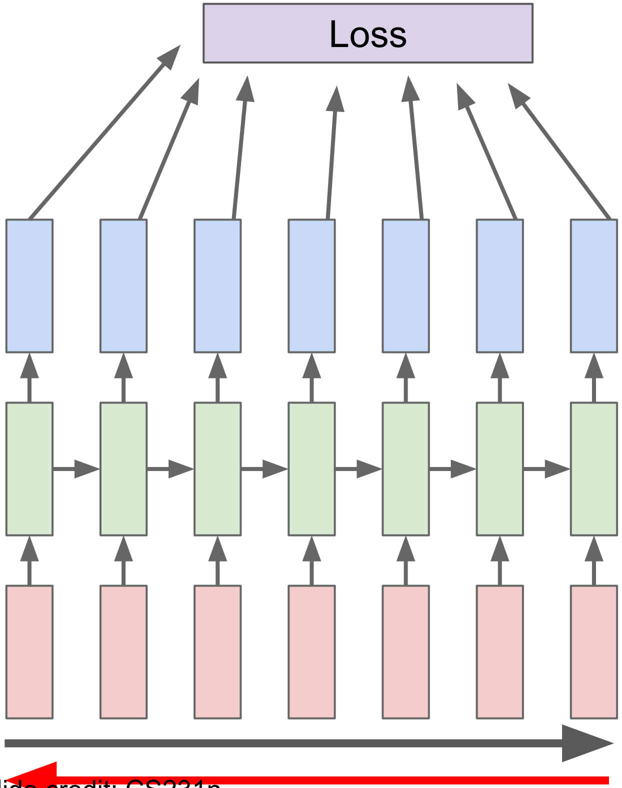
Backpropagation through time

Forward through entire sequence to compute loss, then backward through entire sequence to compute gradient



Slide credit: CS231n

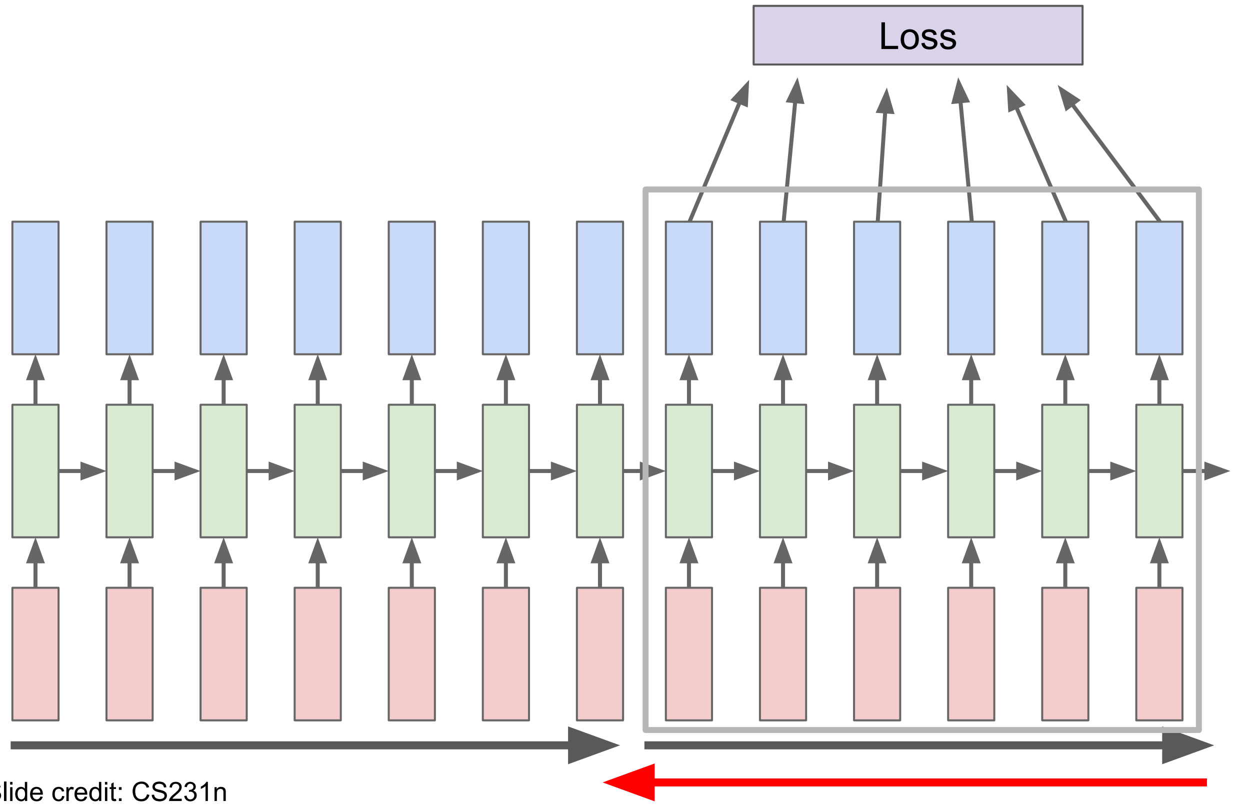
Truncated Backpropagation through time



Run forward and backward through chunks of the sequence instead of whole sequence

Slide credit: CS231n

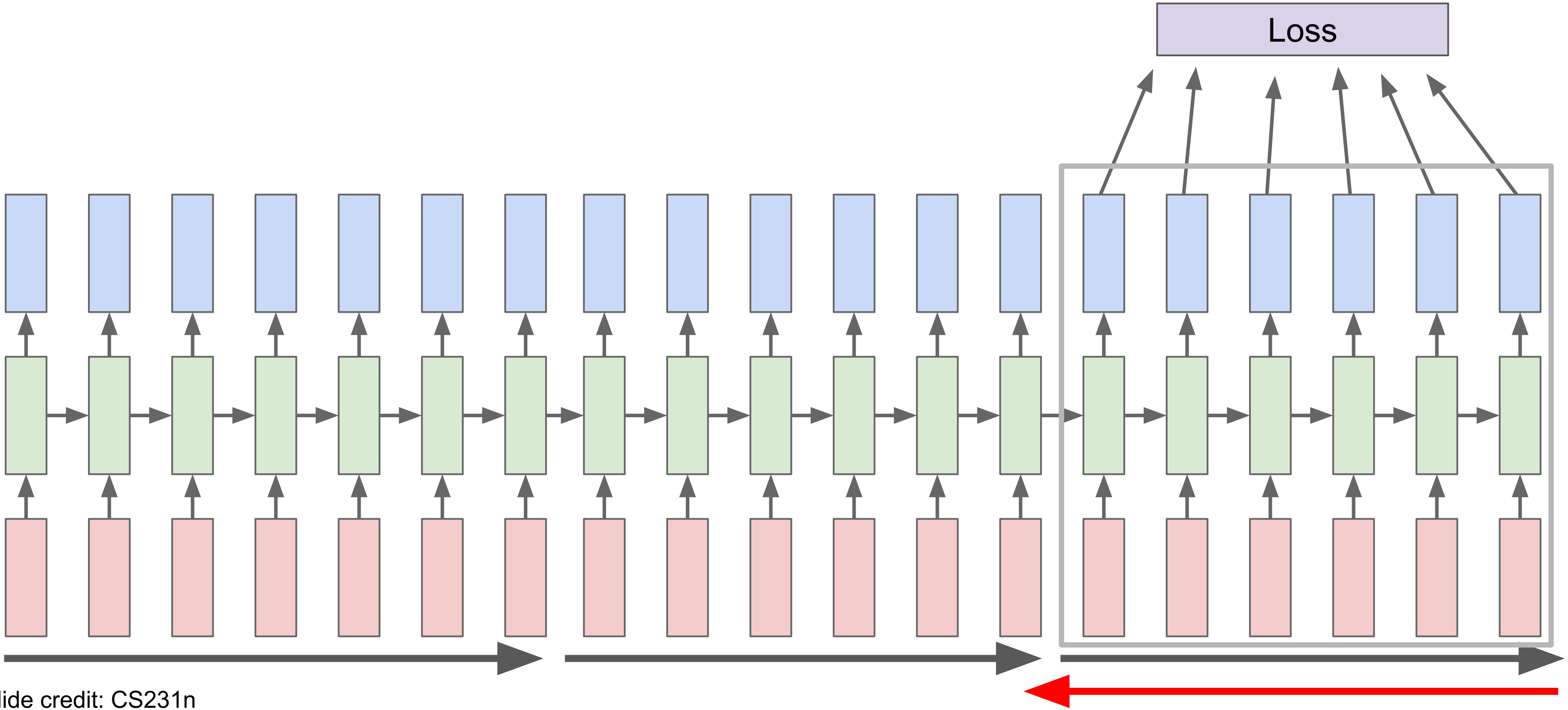
Truncated Backpropagation through time



Carry hidden states forward in time forever, but only backpropagate for some smaller number of steps

Slide credit: CS231n

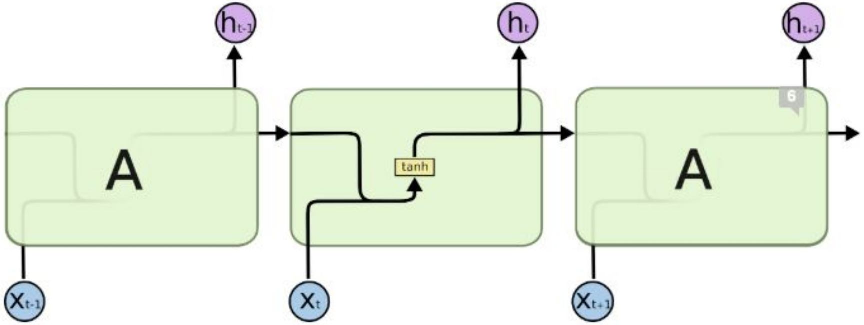
Truncated Backpropagation through time



Slide credit: CS231n

Long Short Term Memory (LSTM) Recurrent Networks

Unrolled Vanilla RNN



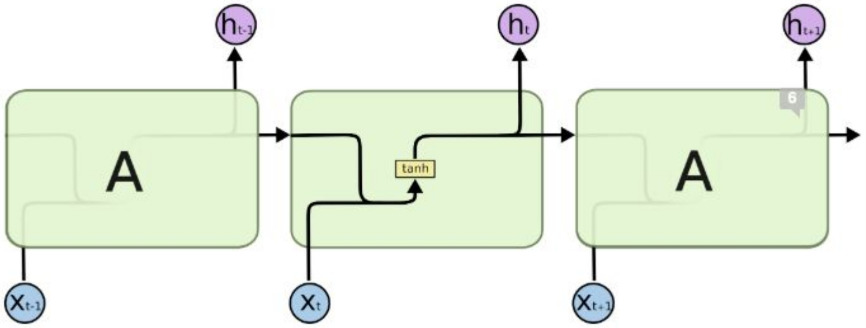
$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$

$$y_t = W_{hy}h_t$$

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

Long Short Term Memory (LSTM) Recurrent Networks

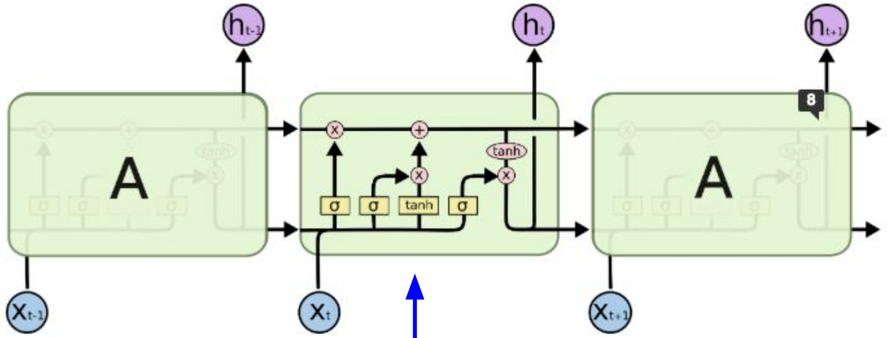
Unrolled Vanilla RNN



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

$$y_t = W_{hy}h_t$$

Unrolled LSTM



Different computation to obtain h_t

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

Long Short Term Memory (LSTM) Recurrent Networks

"Cell state" flows through entire sequence. At each timestep, will be able to modify the cell state.

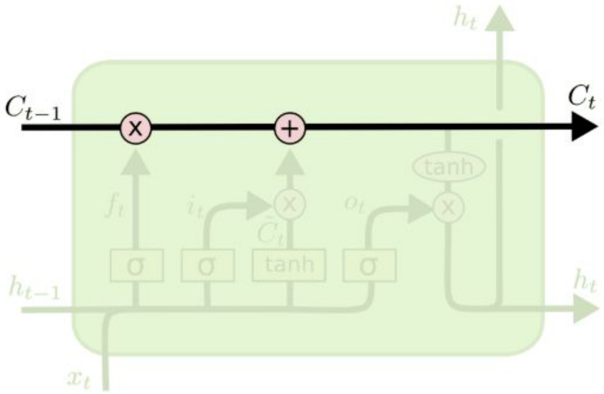
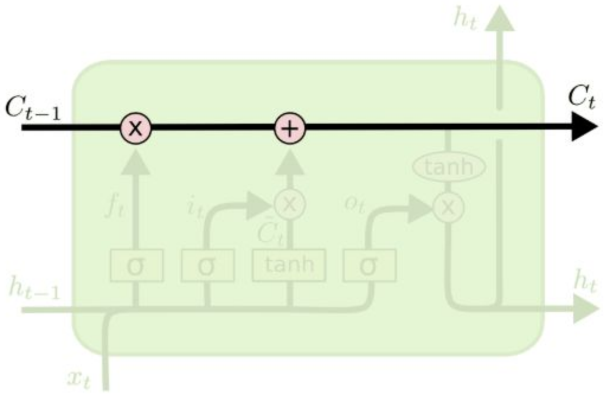


Figure credit: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Long Short Term Memory (LSTM) Recurrent Networks

"Cell state" flows through entire sequence. At each timestep, will be able to modify the cell state.



Gates (sigmoid + elementwise multiplication) control passing of information.
Sigmoid output of 1 = let everything through; output of 0 = let nothing through.

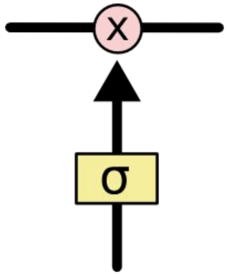
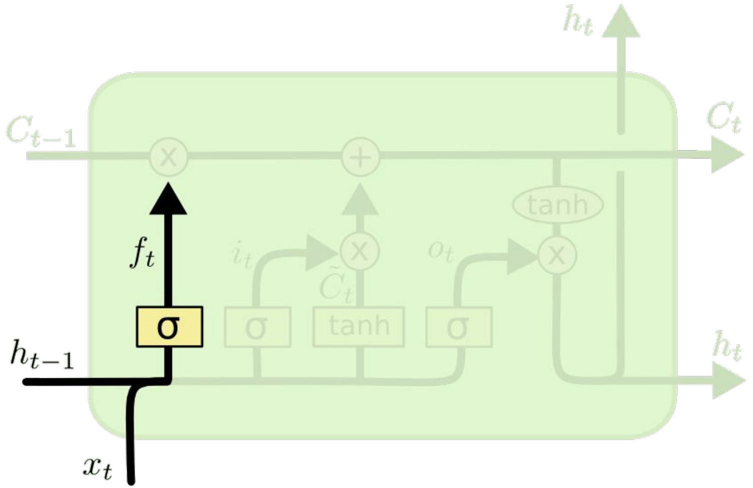


Figure credit: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Long Short Term Memory (LSTM) Recurrent Networks



Fully connected layer

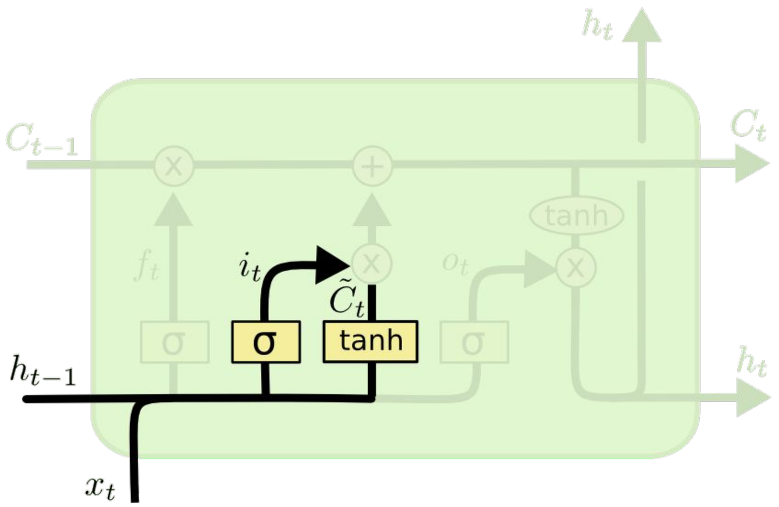
$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

"Forget" gate

Concatenate

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

Long Short Term Memory (LSTM) Recurrent Networks



"Input" gate

Fully connected layer

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$

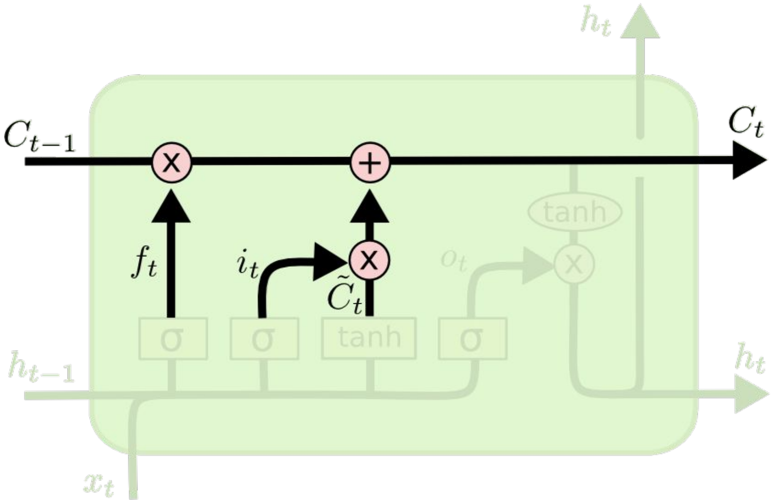
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

New "candidate" values that could be added to modify cell state

Fully connected layer

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

Long Short Term Memory (LSTM) Recurrent Networks



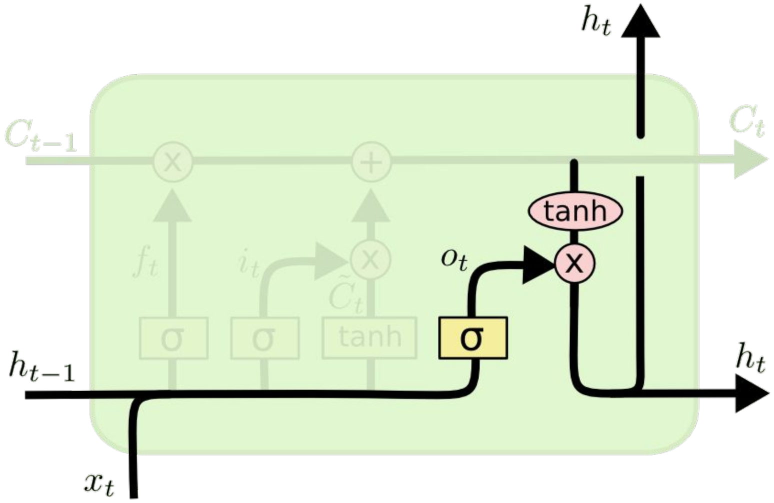
New cell state

$$\underbrace{C_t}_{\text{New cell state}} = \underbrace{f_t}_{\text{Forget gate}} * \underbrace{C_{t-1}}_{\text{Previous cell state}} + \underbrace{i_t}_{\text{Input gate}} * \underbrace{\tilde{C}_t}_{\text{New candidate values}}$$

Forget gate | Previous cell state | Input gate | New candidate values

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

Long Short Term Memory (LSTM) Recurrent Networks



"Output" gate

Fully connected layer

$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

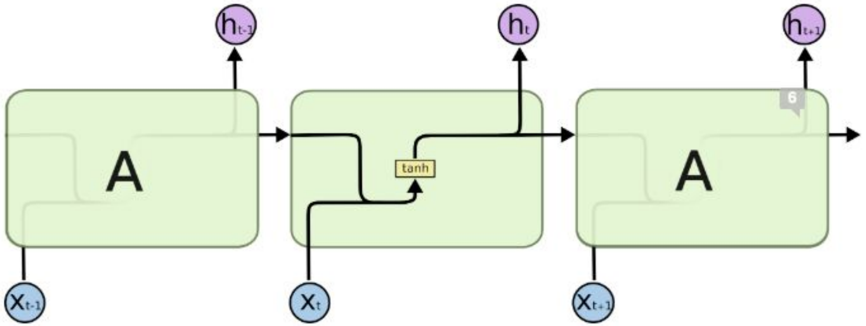
New LSTM layer output h_t

Current cell state

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

Long Short Term Memory (LSTM) Recurrent Networks

Unrolled Vanilla RNN



Unrolled LSTM

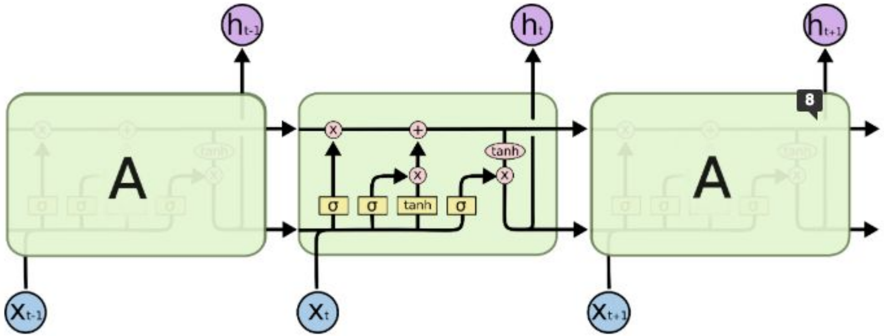
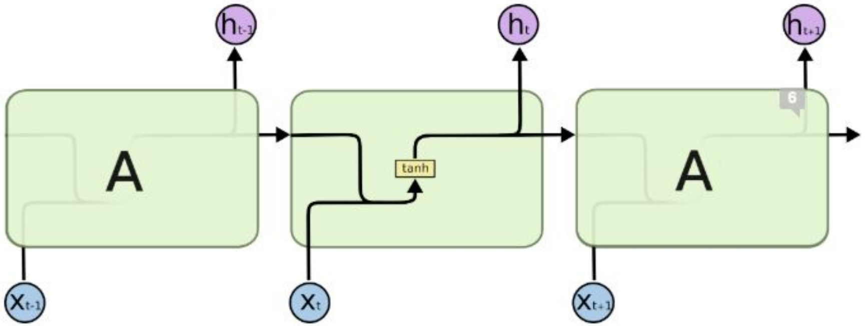


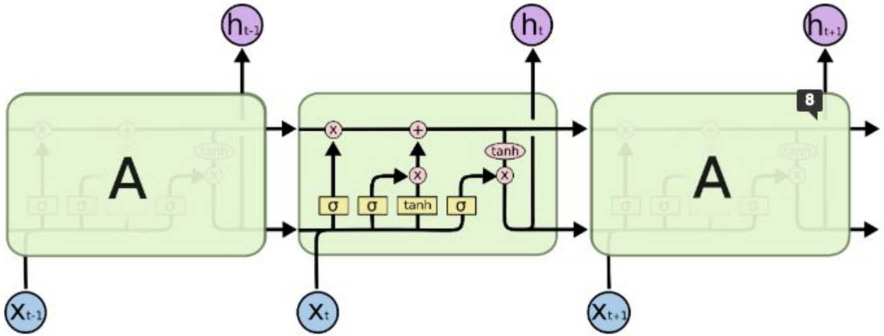
Figure credit: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Long Short Term Memory (LSTM) Recurrent Networks

Unrolled Vanilla RNN



Unrolled LSTM

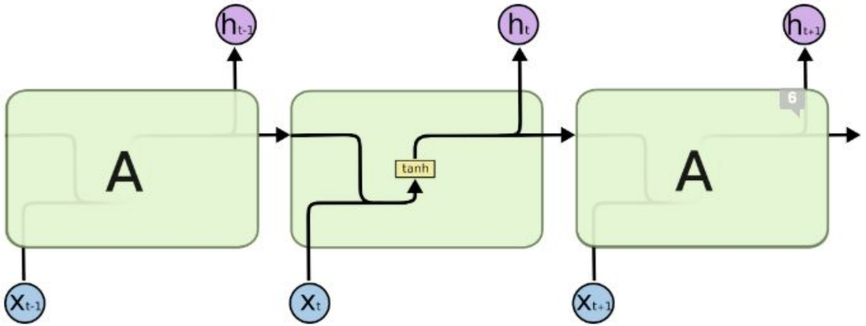


Usage of a "cell state" in the LSTM that is modified through addition allows improved gradient flow through longer sequences.

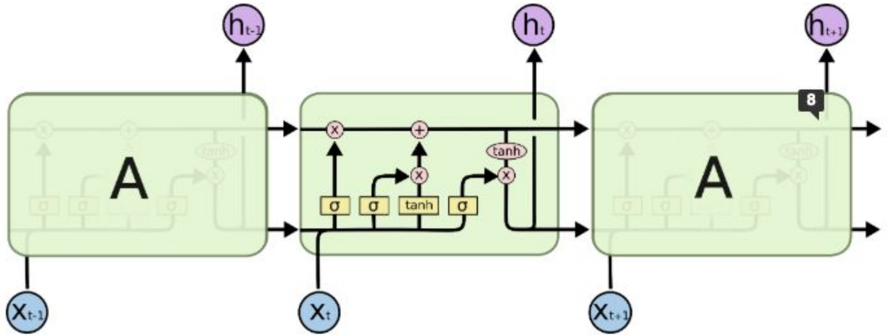
Figure credit: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Long Short Term Memory (LSTM) Recurrent Networks

Unrolled Vanilla RNN



Unrolled LSTM



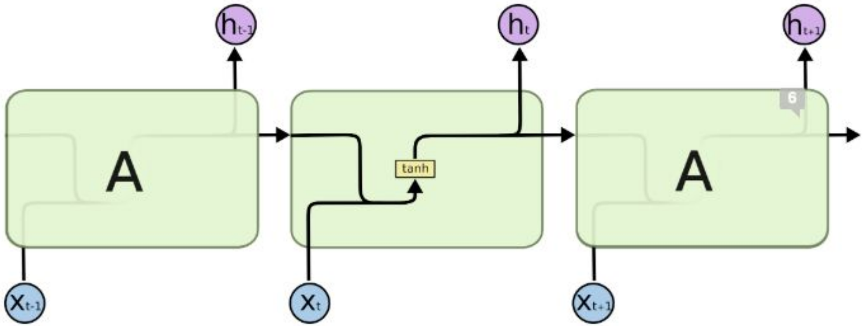
LSTM often used over Vanilla RNN in practice.

Usage of a “cell state” in the LSTM that is modified through addition allows improved gradient flow through longer sequences.

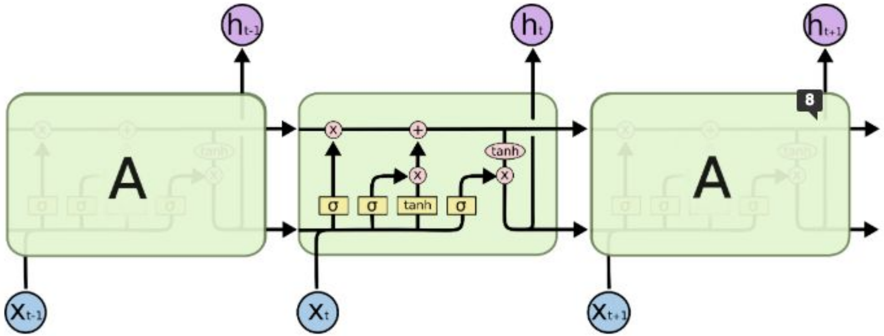
Figure credit: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Long Short Term Memory (LSTM) Recurrent Networks

Unrolled Vanilla RNN



Unrolled LSTM



LSTM often used over Vanilla RNN in practice.

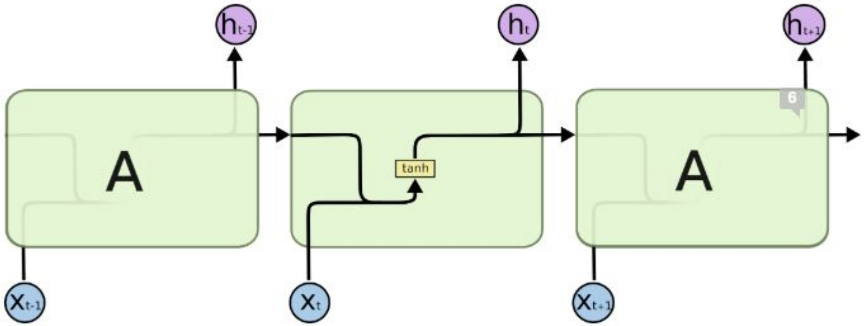
Usage of a “cell state” in the LSTM that is modified through addition allows improved gradient flow through longer sequences.

Will also see other variants e.g. GRUs with different gating operations.

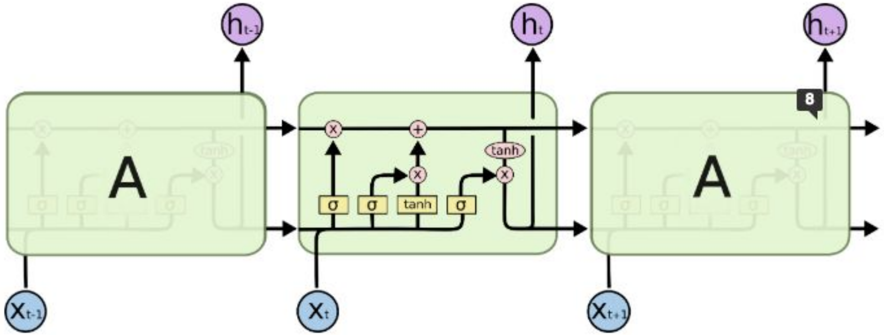
Figure credit: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Long Short Term Memory (LSTM) Recurrent Networks

Unrolled Vanilla RNN



Unrolled LSTM



Can have multi-layer RNNs and LSTMs, where the {h} outputs of one layer form the input sequence for the next layer. One or two layers is common.

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

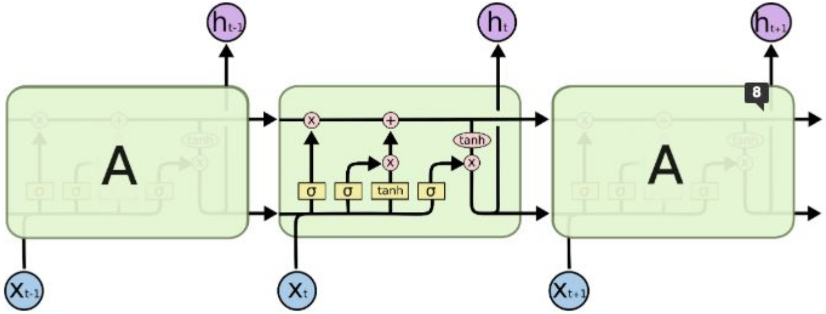
Harutyunyan et al. Multitask learning and benchmarking with clinical time series data. 2019.

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.: in-hospital mortality

- Input: Time-series data for first 48 hours of ICU stay
- Output: binary classification of in-hospital mortality

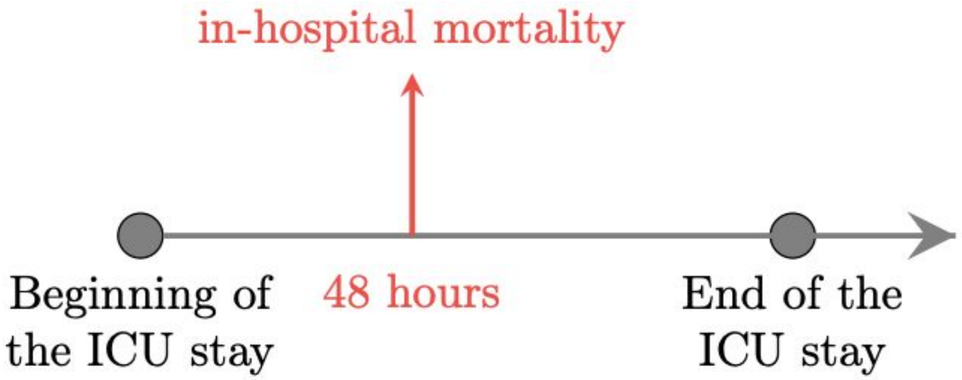


Figure credit: Harutyunyan et al. Multitask learning and benchmarking with clinical time series data. 2019.

Harutyunyan et al.: in-hospital mortality

- Input: Time-series data for first 48 hours of ICU stay
- Output: binary classification of in-hospital mortality

Many-to-one LSTM

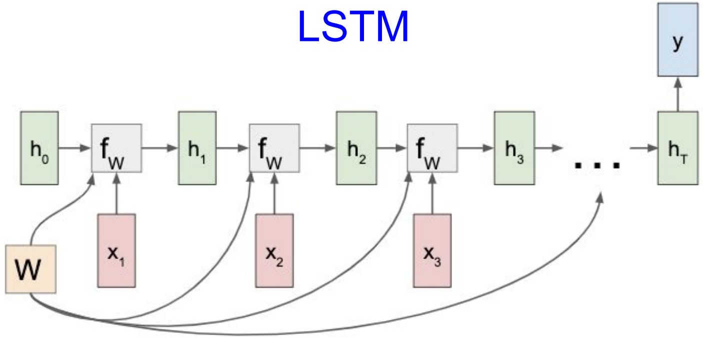


Figure credit: Harutyunyan et al. Multitask learning and benchmarking with clinical time series data. 2019.

Harutyunyan et al.: decompensation

- Input: Time-series data from beginning of stay until prediction time (every hour)
- Output: Binary classification of mortality in the next 24 hours

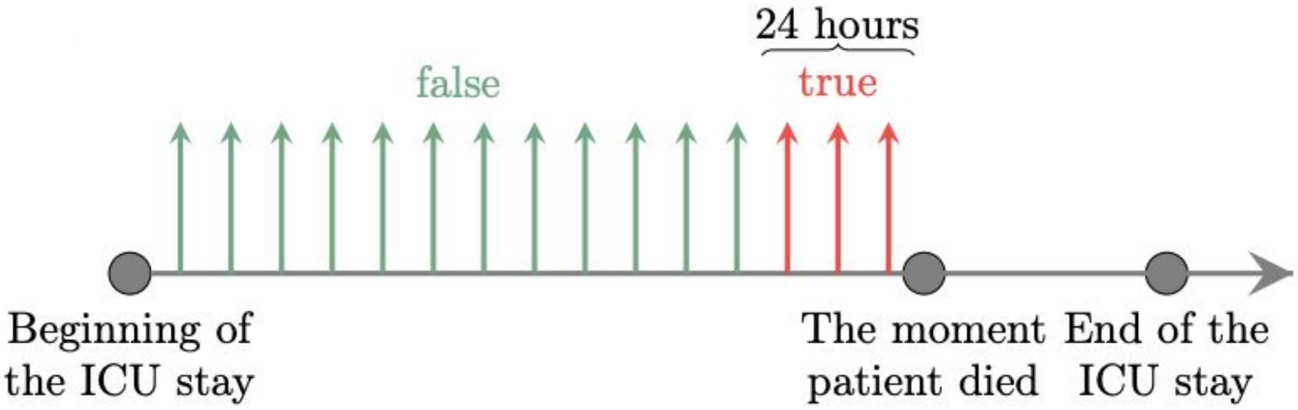
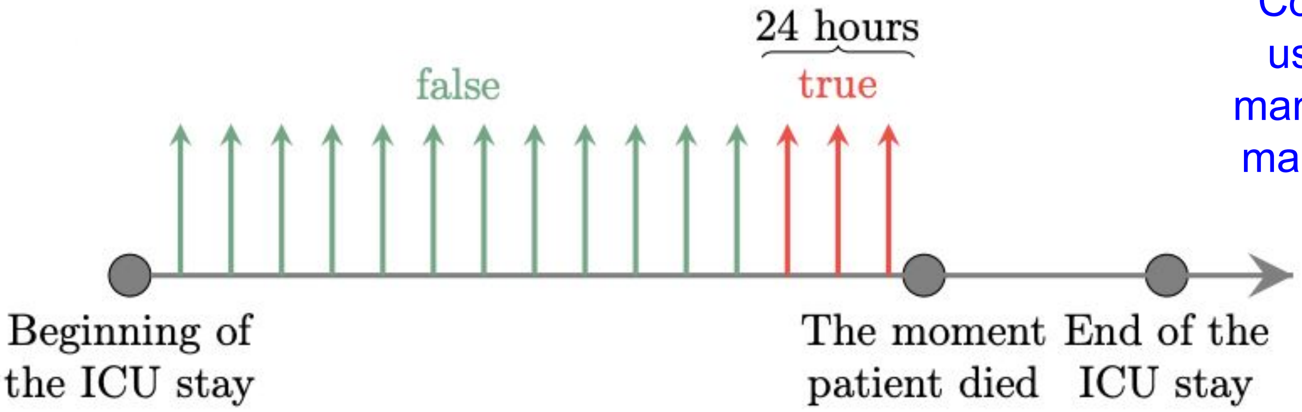


Figure credit: Harutyunyan et al. Multitask learning and benchmarking with clinical time series data. 2019.

Harutyunyan et al.: decompensation

- Input: Time-series data from beginning of stay until prediction time (every hour)
- Output: Binary classification of mortality in the next 24 hours

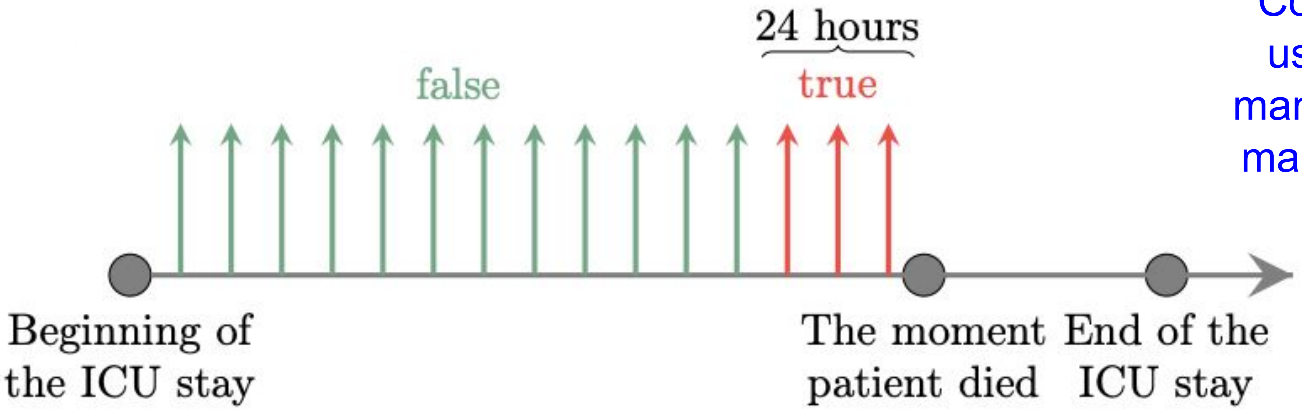


Could model using either many-to-one or many-to-many LSTM

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

Harutyunyan et al.: decompensation

- Input: Time-series data from beginning of stay until prediction time (every hour)
- Output: Binary classification of mortality in the next 24 hours



Could model using either many-to-one or many-to-many LSTM

Closely related to in-hospital mortality

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

Harutyunyan et al.: length-of-stay

- Input: Time-series data from beginning of stay until prediction time (every hour)
- Output: remaining time spent in ICU. Model as classification problem: ICU stays < 1 day, each of 7 days, between 1-2 weeks, > 2 weeks

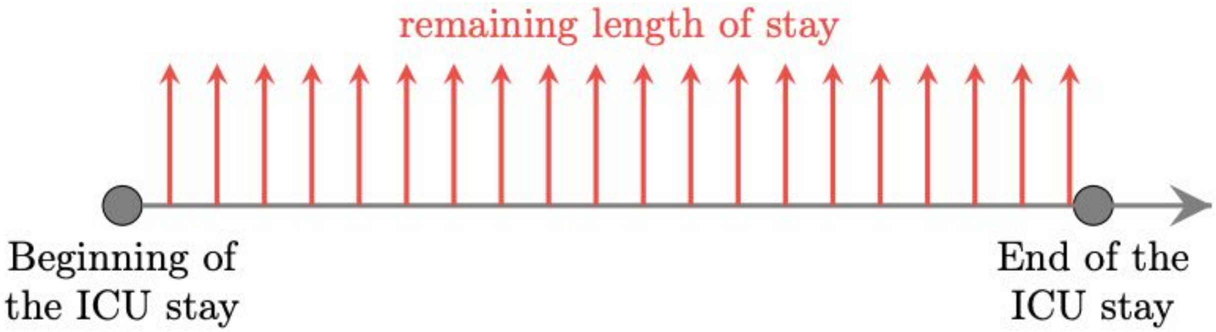
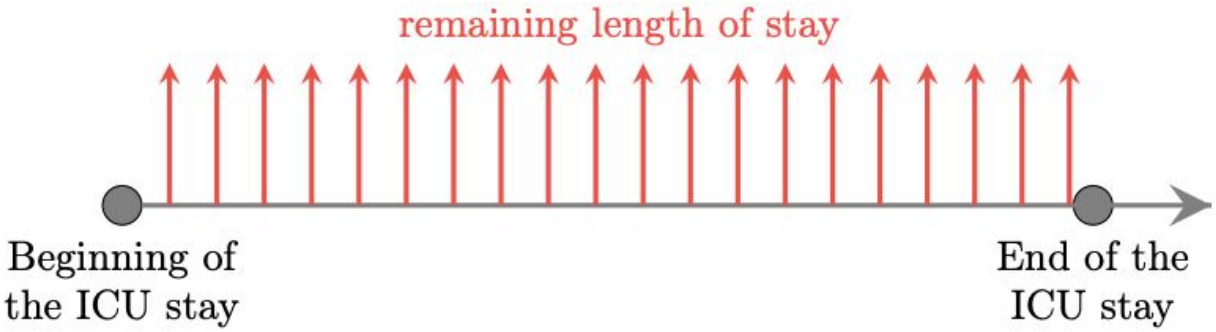


Figure credit: Harutyunyan et al. Multitask learning and benchmarking with clinical time series data. 2019.

Harutyunyan et al.: length-of-stay

- Input: Time-series data from beginning of stay until prediction time (every hour)
- Output: remaining time spent in ICU. Model as classification problem: ICU stays < 1 day, each of 7 days, between 1-2 weeks, > 2 weeks



Can model problem in different ways, e.g. directly regress LOS value, or predict meaningful category of extended LOS (>7 days)

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

Harutyunyan et al.: phenotypes

- Input: Time-series data corresponding to entire ICU stay
- Output: Multilabel classification of the presence of 25 acute care conditions (merged from ICD codes) in stay record

Phenotype
Acute and unspecified renal failure
Acute cerebrovascular disease
Acute myocardial infarction
Cardiac dysrhythmias
Chronic kidney disease
Chronic obstructive pulmonary disease
Complications of surgical/medical care
Conduction disorders
Congestive heart failure; nonhypertensive
Coronary atherosclerosis and related
Diabetes mellitus with complications
Diabetes mellitus without complication
Disorders of lipid metabolism

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

Harutyunyan et al.: phenotypes

- Input: Time-series data corresponding to entire ICU stay
- Output: Multilabel classification of the presence of 25 acute care conditions (merged from ICD codes) in stay record

Q: Why do we formulate this as a multi-label classification task?

Phenotype
Acute and unspecified renal failure
Acute cerebrovascular disease
Acute myocardial infarction
Cardiac dysrhythmias
Chronic kidney disease
Chronic obstructive pulmonary disease
Complications of surgical/medical care
Conduction disorders
Congestive heart failure; nonhypertensive
Coronary atherosclerosis and related
Diabetes mellitus with complications
Diabetes mellitus without complication
Disorders of lipid metabolism

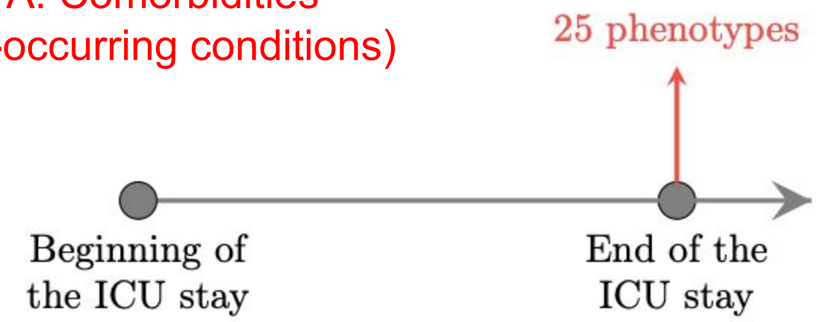
Figure credit: Harutyunyan et al. Multitask learning and benchmarking with clinical time series data. 2019.

Harutyunyan et al.: phenotypes

- Input: Time-series data corresponding to entire ICU stay
- Output: Multilabel classification of the presence of 25 acute care conditions (merged from ICD codes) in stay record

Q: Why do we formulate this as a multi-label classification task?

A: Comorbidities (co-occurring conditions)



Phenotype
Acute and unspecified renal failure
Acute cerebrovascular disease
Acute myocardial infarction
Cardiac dysrhythmias
Chronic kidney disease
Chronic obstructive pulmonary disease
Complications of surgical/medical care
Conduction disorders
Congestive heart failure; nonhypertensive
Coronary atherosclerosis and related
Diabetes mellitus with complications
Diabetes mellitus without complication
Disorders of lipid metabolism

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

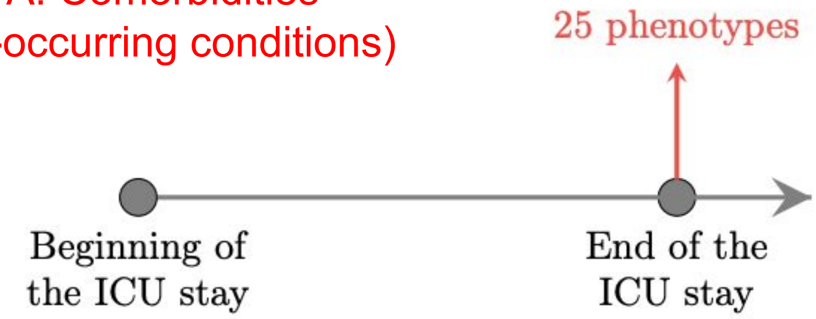
Harutyunyan et al.: phenotypes

- Input: Time-series data corresponding to entire ICU stay
- Output: Multilabel classification of the presence of 25 acute care conditions (merged from ICD codes) in stay record

Q: Why do we formulate this as a multi-label classification task?

A: Comorbidities (co-occurring conditions)

Q: What loss function should we use?



Phenotype
Acute and unspecified renal failure
Acute cerebrovascular disease
Acute myocardial infarction
Cardiac dysrhythmias
Chronic kidney disease
Chronic obstructive pulmonary disease
Complications of surgical/medical care
Conduction disorders
Congestive heart failure; nonhypertensive
Coronary atherosclerosis and related
Diabetes mellitus with complications
Diabetes mellitus without complication
Disorders of lipid metabolism

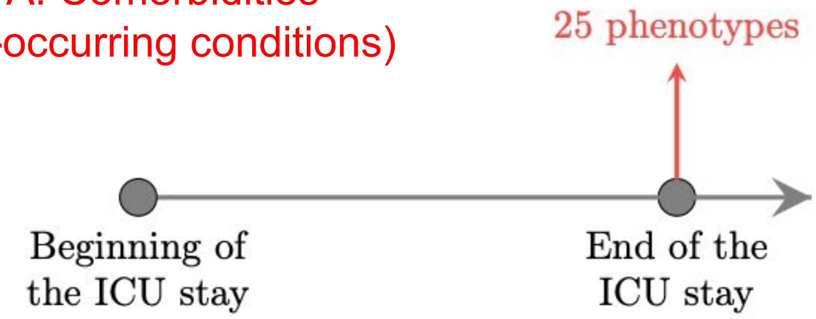
Figure credit: Harutyunyan et al. Multitask learning and benchmarking with clinical time series data. 2019.

Harutyunyan et al.: phenotypes

- Input: Time-series data corresponding to entire ICU stay
- Output: Multilabel classification of the presence of 25 acute care conditions (merged from ICD codes) in stay record

Q: Why do we formulate this as a multi-label classification task?

A: Comorbidities (co-occurring conditions)



Phenotype
Acute and unspecified renal failure
Acute cerebrovascular disease
Acute myocardial infarction
Cardiac dysrhythmias
Chronic kidney disease
Chronic obstructive pulmonary disease
Complications of surgical/medical care
Conduction disorders
Congestive heart failure; nonhypertensive
Coronary atherosclerosis and related
Diabetes mellitus with complications
Diabetes mellitus without complication
Disorders of lipid metabolism

Q: What loss function should we use?

A: Multiple binary cross-entropy losses

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

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.

Summary:

- Introduction to EHRs
- EHR prediction tasks
- Recurrent neural networks and LSTMs

Next:

- More on EHR data
- More on feature representations and model interpretability