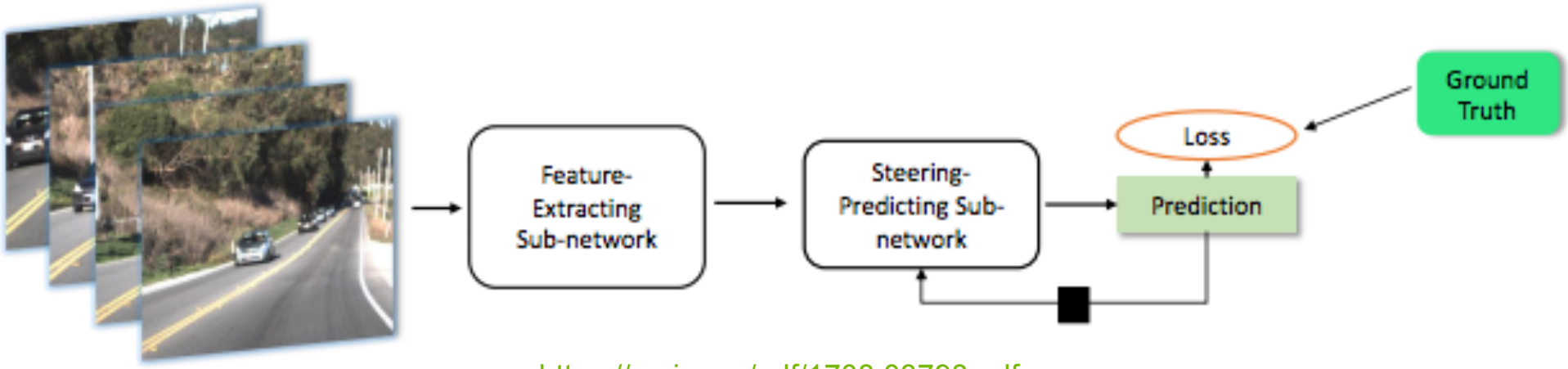


https://arxiv.org/pdf/1708.03798.pdf

eep Sequence Learning

Urbain, PhD
in@msoe.edu

edIn https://www.linkedin.com/in/jayurbain/
ub https://github.com/jayurbain, https://github.com/jayurbain/DeepSequenceLearningIntro
ching http://jayurbain.com/msoe/index.html

Prologue

Traditional machine learning methods require significant feature engineering to help capture the underlying concepts and relations in data.

Feature engineering is especially common in multi-variate sequence learning tas to capture spatial or temporal relations in data,

Deep learning methods can learn hierarchical representations of concepts as par the learning process.

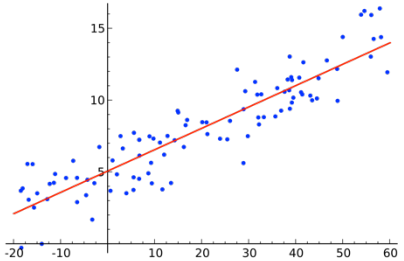
Can we learn effective end-to-end models using deep learning for sequence learning applications?

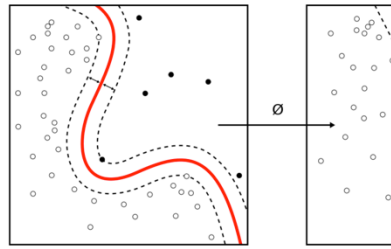
Difference in Workflow

Classic Machine Learning [1990 : now]

ut → Hand Designed **Features** → Model / Mapping → Output

Examples [Regression and SVMs

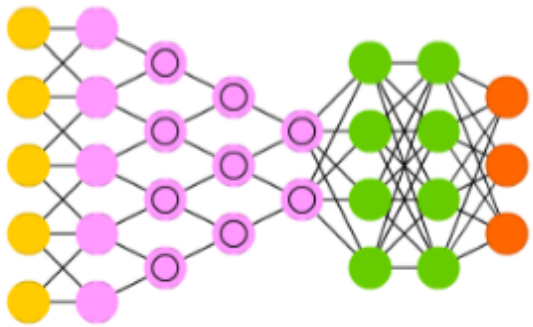




Deep/End-to-End Learning [2012 : now]

→ Simple Features → Complex Features → Model/ Mapping → Output

Example [Conv Net]

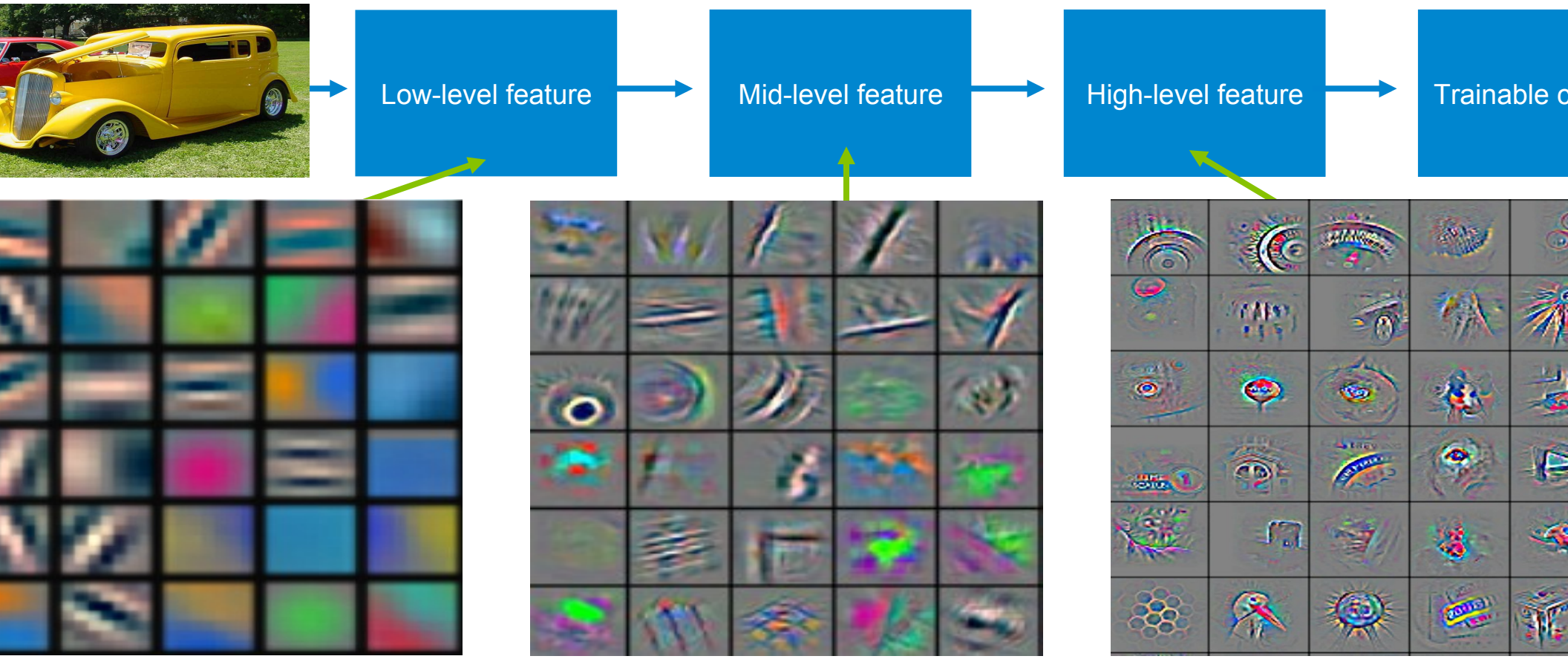


ine learning workflow shifts from engineering features for “shallow” models

chitecting deep learning models with the ability to learn hierarchical representations of featu

eep learning = learning hierarchical epresentations

deep if it has more than one stage of non-linear feature transformations



sualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

Outline

What is a sequence?

Applications

Traditional ML methods for sequences

Deep learning for sequences

RNN Models

ConvNet Models

Hybrid Models

Coding example

Future directions

Sequence Learning

quence data

Data with a sequential dependency across space or time.

Not IID.

quence prediction

Use historical sequence data, predict the next value or values in the sequence.

Example: predict next word in a sentence, future price in a time series of stock prices.

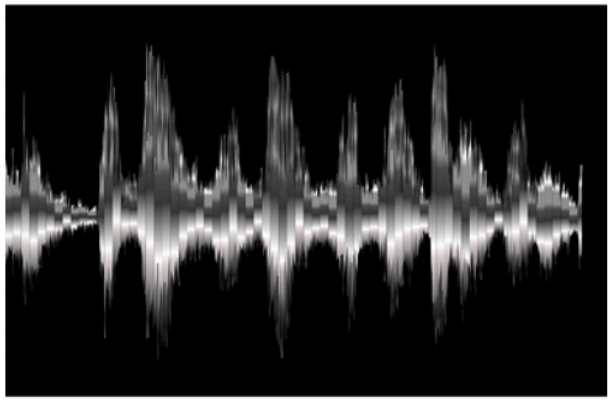
quence classification

Using some sequence of data over space or time, predict a category for the sequence.

Example: activity recognition, arrhythmia classification, named entity recognitio text classification, text summarization.

Sequence Data

xt, video, audio:



me series: finance, industry, medicine

Sequence classification: patient de-identification

https://cis.ctsi.mcw.edu/deid/

jay.urbain@gmail.com, born December 6, 2156 is an elderly caucasian male suffering from illusions of grandeur and LBP. He is married to Kimberly Urbain, looking. Patient father, Francis Urbain has a history of CAD and DM. Jay has been prescribed meloxicam, and venti americano. He lives at 9050 N. Tennys WI with his wife and golden retriever Mel. You can reach him at 414-745-5102.

Pretty Print

ts:

RSON], [xxx@xxx.xxx] , born [12_16_2156] is an elderly caucasian male suffering from illusions of grandeur and LBP. He is married to [PERSON] [PERSON], who is much better looking RSON] has a history of CAD and DM. [PERSON] has been prescribed meloxicam, and venti americano. He lives at [xxxxx x. xxxxx] Dr., Disturbia, WI with his wife and golden retriever [t [xxx_xxx_xxxx] .

Sequence classification: medical named entity recognition

https://cis.ctsi.mcw.edu/nlp/

Parsed results:

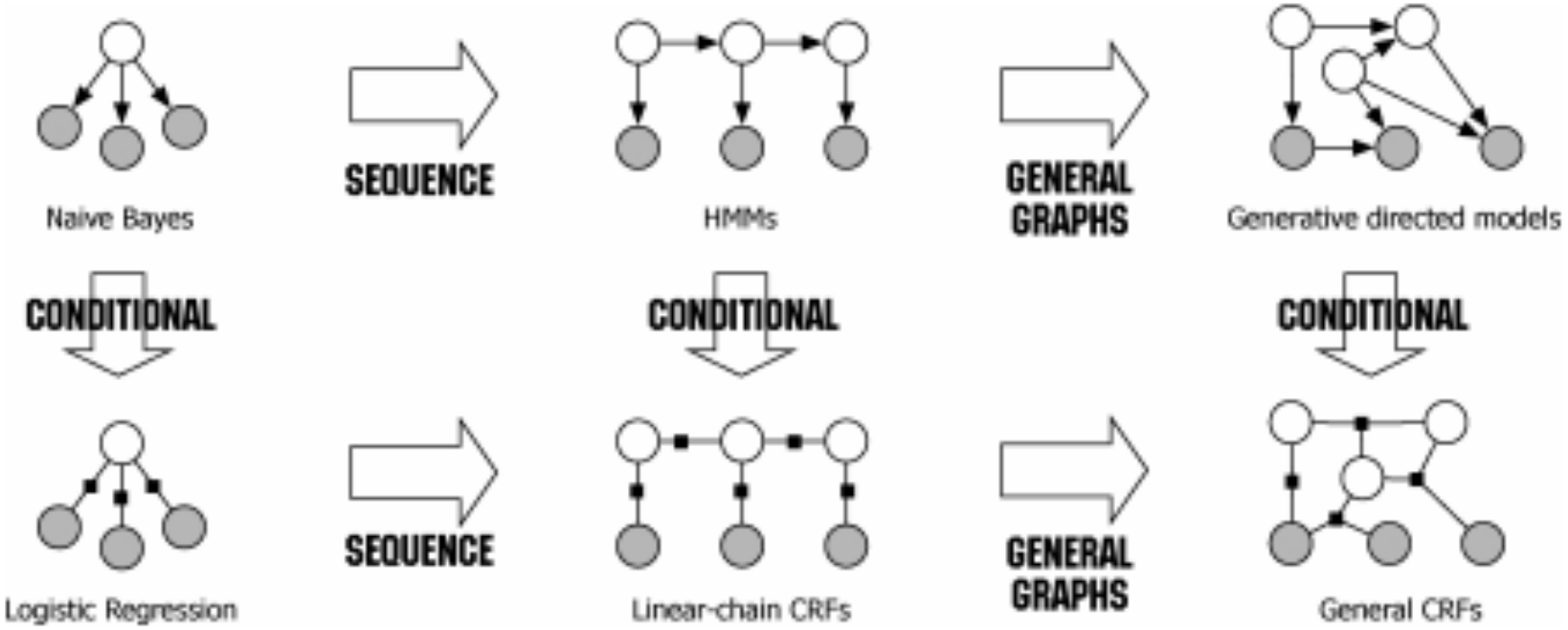
```
SENTENCE:  Jay Urbain is  an elderly caucasian male suffering from illusions of grandeur and low back   pain  .
           NNP  NNP   VBZ DT   JJ       JJ      NN     VBG     IN     NNS    IN    NN    CC  JJ   NN     NN
                                                 |=======|      |=======|    |======|               |======|
                                                   Event        Disorder      Event                 Finding
                                                                C0020903                           C0030193
                                                                                                |==========|
                                                                                                  Finding
                                                                                                  C0004604
                                                                                              |==============|
                                                                                                  Finding
                                                                                                  C0024031


SENTENCE:  Patient has a  family history  of   CAD    and DM.
             NN    VBZ DT   NN      NN    IN    NN    CC  NN
                              |======|    |======|
                              Finding     Disorder
                              C0262926    C1956346
                         |============|
                             Finding
                             C0241889
TLINKS:    history CONTAINS CAD

SENTENCE:  Prescribed meloxicam, and venti americano.
              VBN        NN      CC   JJ      NN
                       |=======|           |=======|
                         Drug                Event
                       C0083381
```

Traditional NLP Sequence Models: HMM, MEMM, CRF

tp://homepages.inf.ed.ac.uk/csutton/publications/crftut-fnt.pdf



Traditional Models: HMM, MEMM, CRF

tp://homepages.inf.ed.ac.uk/csutton/publications/crftut-fnt.pdf

near chain CRF for sequence class.

on 2.2. Let Y, X be random vectors, $\theta = \{\theta_k\} \in \Re^K$ be a er vector, and $\mathcal{F} = \{f_k(y, y', \mathbf{x}_t)\}_{k=1}^K$ be a set of real-valued unctions. Then a *linear-chain conditional random field* is a ion $p(\mathbf{y}|\mathbf{x})$ that takes the form:

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{t=1}^{T} \exp\left\{ \sum_{k=1}^{K} \theta_k f_k(y_t, y_{t-1}, \mathbf{x}_t) \right\}, \qquad (2.18)$$

$(\mathbf{x})$ is an input-dependent normalization function

$$Z(\mathbf{x}) = \sum_{\mathbf{y}} \prod_{t=1}^{T} \exp\left\{ \sum_{k=1}^{K} \theta_k f_k(y_t, y_{t-1}, \mathbf{x}_t) \right\}. \qquad (2.19)$$

Named entity CRF features

W=v	$w_t = v$
T=j	part-of-speech tag for w_t is j (as determined by an automatic tagger)
P=I−j	w_t is part of a phrase with syntactic type j (as determined by an automatic chunker)
Capitalized	w_t matches `[A-Z][a-z]+`
Allcaps	w_t matches `[A-Z][A-Z]+`
EndsInDot	w_t matches `[^\.]+.*\.`
	w_t contains a dash
	w_t matches `[A-Z]+[a-z]+[A-Z]+[a-z]`
Acro	w_t matches `[A-Z][A-Z\\.]*\\.[A-Z\\.]*`
Stopword	w_t appears in a hand-built list of stop words
CountryCapital	w_t appears in list of capitals of countries
⋮	many other lexicons and regular expressions

$q_k(\mathbf{x}, t + \delta)$ for all k and $\delta \in [-1, 1]$

CRF libraries

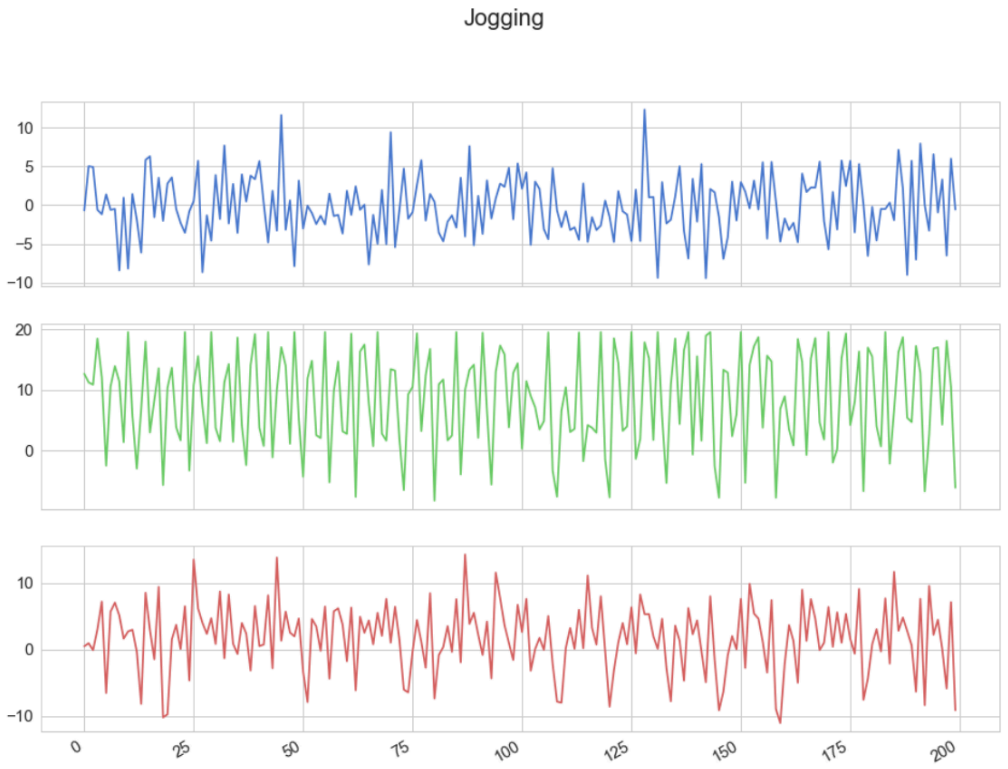
anford has a very nice Java implementation and pre-trained models for tradition
ntities, i.e., person, location, organization etc.

CRF++	http://crfpp.sourceforge.net/
MALLET	http://mallet.cs.umass.edu/
GRMM	http://mallet.cs.umass.edu/grmm/
CRFSuite	http://www.chokkan.org/software/crfsuite/
FACTORIE	http://www.factorie.cc

Sequence classification: activity recognition

mporal sensor analysis and activity tracking

user	activity	timestamp	x-axis	y-axis	z-axis
33	Jogging	49105962326000	-0.694638	12.680544	0.503953
33	Jogging	49106062271000	5.012288	11.264028	0.953424
33	Jogging	49106112167000	4.903325	10.882658	-0.081722
33	Jogging	49106222305000	-0.612916	18.496431	3.023717
33	Jogging	49106332290000	-1.184970	12.108489	7.205164



Traditional models: activity recognition

tp://www.cis.fordham.edu/wisdm/public_files/sensorKDD-2010.pdf

rage[3]: Average acceleration (for each axis)

ndard Deviation[3]: Standard deviation (for each axis)

rage Absolute Difference[3]: Average absolute erence between the value of each of the 200 readings in the ED and the mean value over those 200 values each axis)

rage Resultant Acceleration[1]: Average of the square s of the sum of the values of each axis squared $^2 + y_i^2 + z_i^2$) over the ED

e Between Peaks[3]: Time in milliseconds between ks in the sinusoidal waves associated with most vities (for each axis)

ned Distribution[30]: We determine the range of values each axis (maximum – minimum), divide this range into equal sized bins, and then record what fraction of the values fell within each of the bins.

Table 2: Accuracies of Activity Recognition

	% of Records Correctly Predicted			
	J48	**Logistic Regression**	**Multilayer Perceptron**	**Straw Man**
Walking	89.9	**93.6**	91.7	37.2
Jogging	96.5	98.0	**98.3**	29.2
Upstairs	59.3	27.5	**61.5**	12.2
Downstairs	**55.5**	12.3	44.3	10.0
Sitting	**95.7**	92.2	95.0	6.4
Standing	**93.3**	87.0	91.9	5.0
Overall	85.1	78.1	**91.7**	37.2

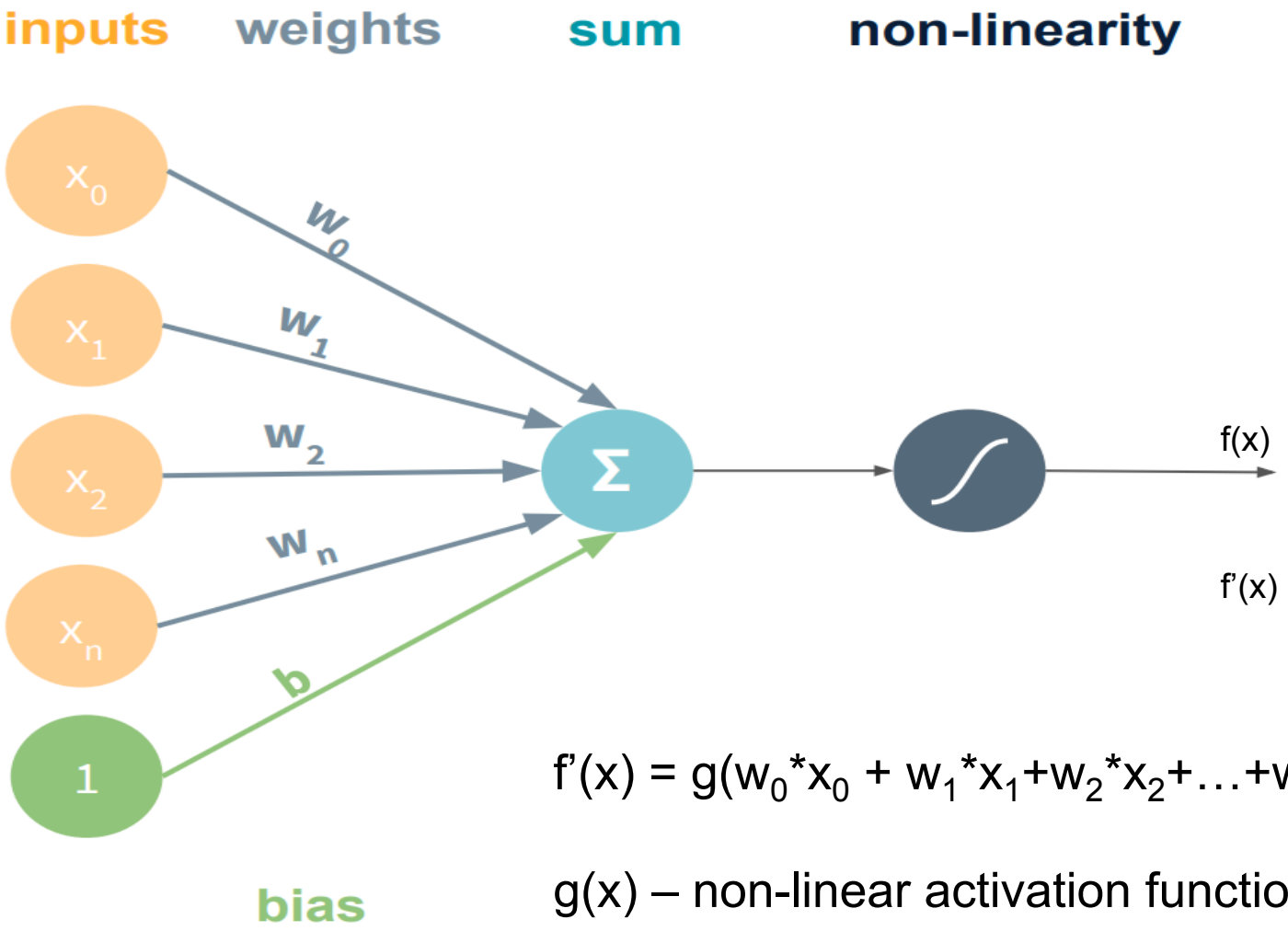
	user	activity	timestamp	x-axis	y-axis	z-
0	33	Jogging	49105962326000	-0.694638	12.680544	0.50
1	33	Jogging	49106062271000	5.012288	11.264028	0.95
2	33	Jogging	49106112167000	4.903325	10.882658	-0.08
3	33	Jogging	49106222305000	-0.612916	18.496431	3.02
4	33	Jogging	49106332290000	-1.184970	12.108489	7.20

Traditional models: activity recognition

te: Improved features using traditional machine learning models (SVM, RF, LR) c
ield an accuracy in the ~92% range (Quihao Jin, Jay Urbain):

3-sec mean
5-sec mean
Median filtering
Acceleration axis deltas
FFT – Fast Fourier Transform

Deep Learning - Perceptron



$f'(x) = g(w_0*x_0 + w_1*x_1+w_2*x_2+\ldots+w_n*x_n + b)$

g(x) – non-linear activation function

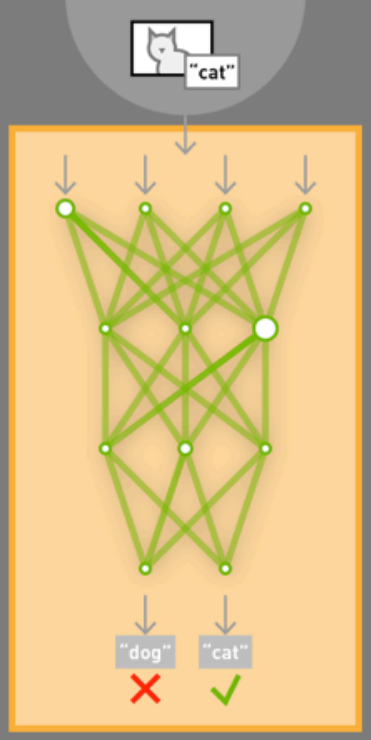
DEEP LEARNING

TRAINING

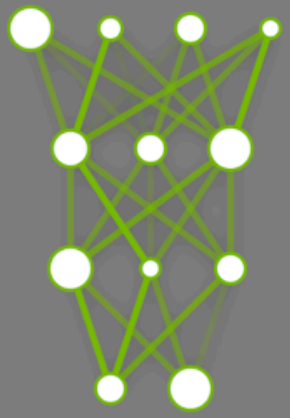
Learning a new capability
from existing data

Deep Learning
Framework

TRAINING
DATASET



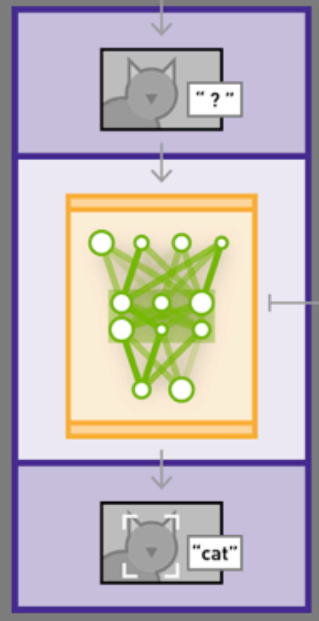
Trained Model
New Capability



INFERENCE

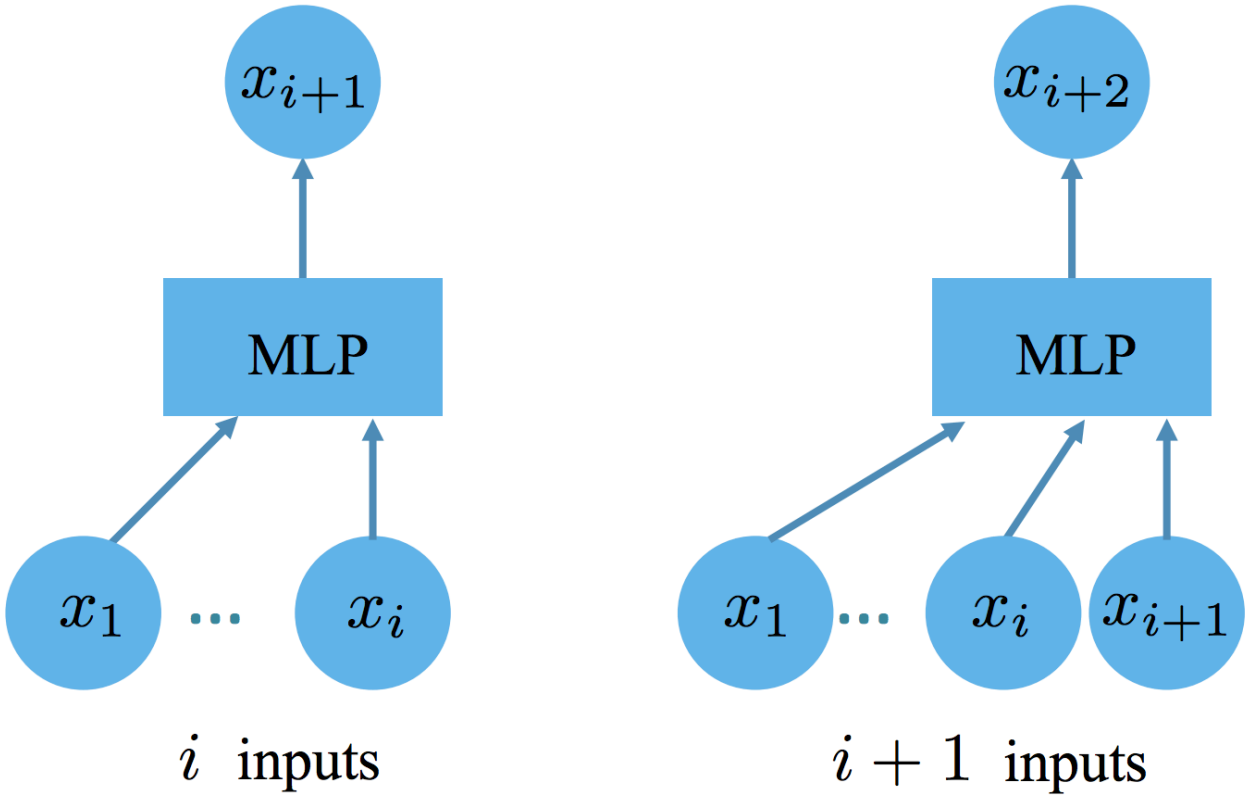
Applying this capability
to new data

NEW
DATA



Why not MLP for sequences?

oblem 1: arbitrary length of sequences:

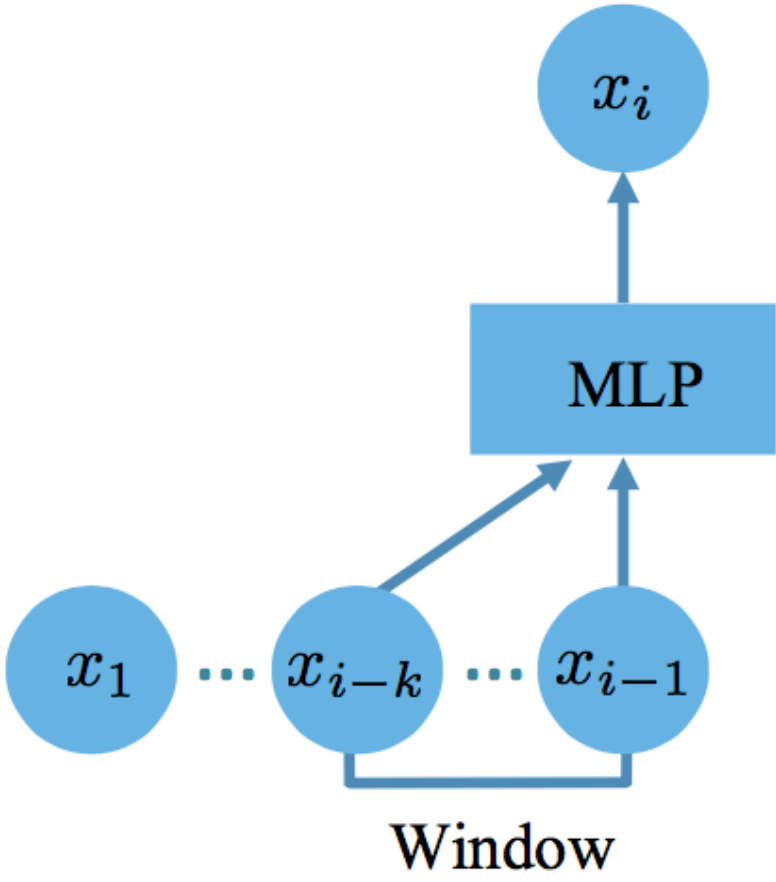


Why not MLP for sequences?

oblem 1: arbitrary length of sequences:

n use a **window of fixed size**:
Just a heuristic – need to choose window size.
Some tasks require wide window and
therefore there is a problem with the large
number of parameters (weights)

estion
How many weights are there in the first layer
of the MLP?
Hidden layer neurons: 100
Input window width: 100
Word embeddings size: 100



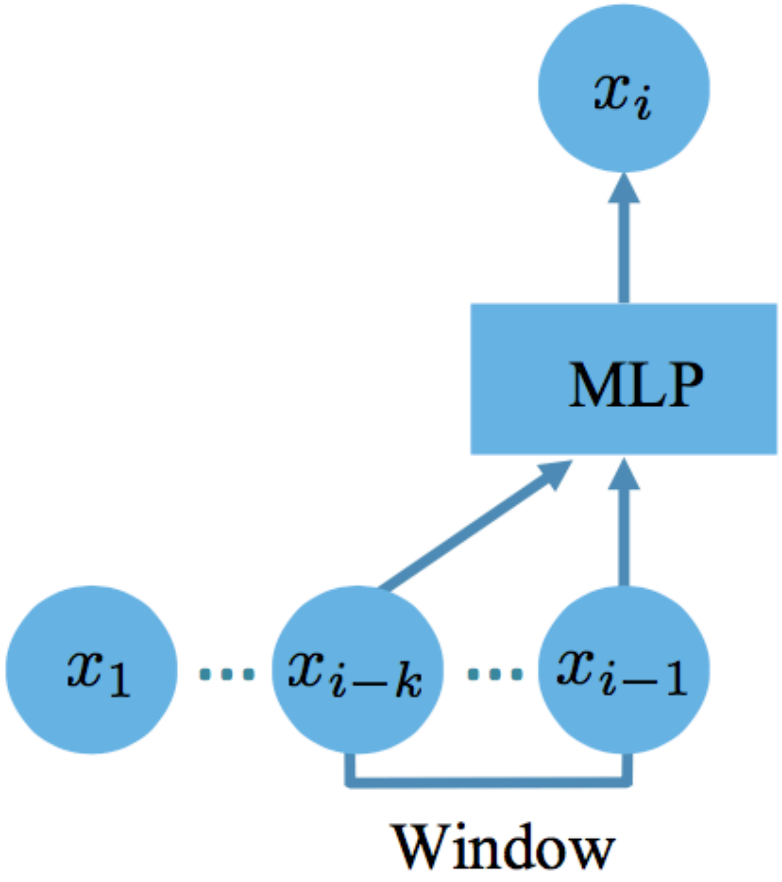
Why not MLP for sequences?

oblem 1: arbitrary length of sequences:

n use a window of fixed size:
Just a heuristic – need to choose window size.
Some tasks require wide window and therefore there is a problem with the large number of parameters (weights)

estion
How many weights are there in the first layer of the MLP?
Hidden layer neurons: 100
Input window width: 100
Word embeddings size: 100

> 100*100*100+100
00100



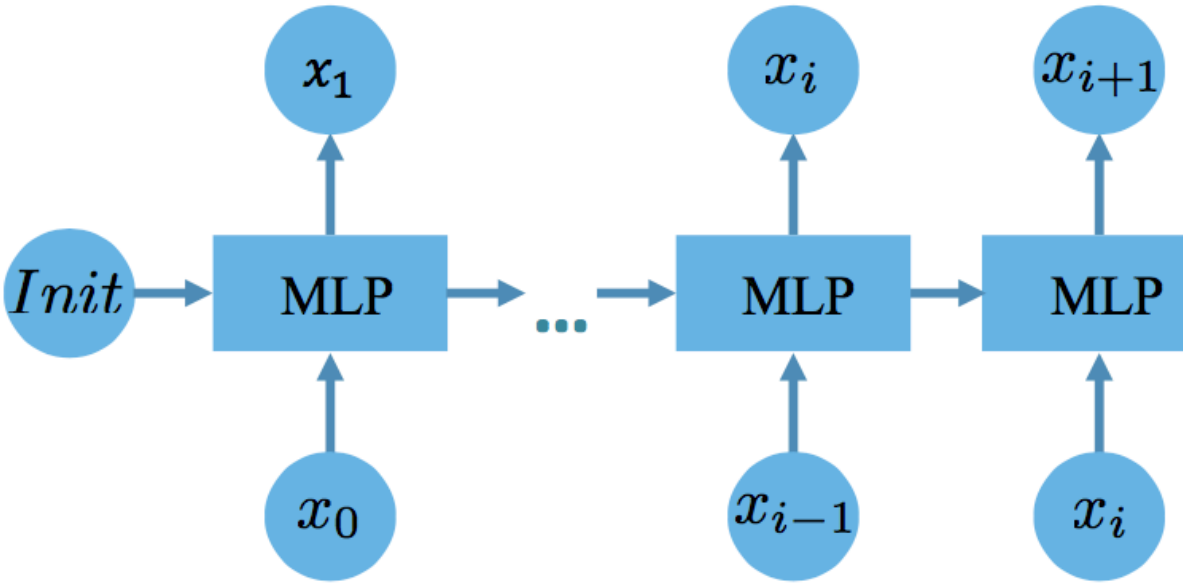
Consider recurrent architecture

xes problem 1: arbitrary length quences.

- Fixed number of inputs at each time step.
- At the first step we use some initial vector as an input from previous time step.

oblem #2: number of rameters to train.

- ution: All the parameters of an P are shared across the different ne steps so we need a much aller number of parameters.
- Only need 20100 parameters
- Translation invariance

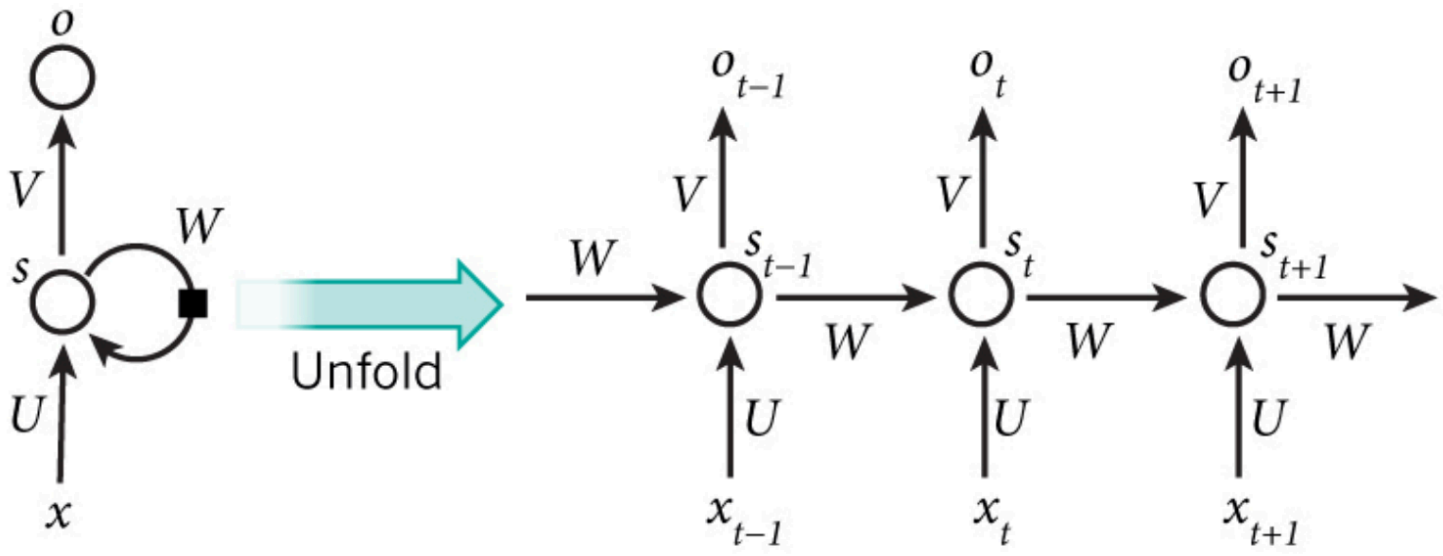


olded/unfolded: Recurrent Neural Networ

RNNs have connections between units along a sequence.

RNN's use the hidden state from the last time step and the input at the current step to make predictions.

Allow RNN's can capture dynamic temporal behavior for a time sequence.



SKIP - Long Short Term Memory (LSTM)

NN's have trouble with long-term dependencies.

ong chain of differentiation when back propagating can cause vanishing gradients

ompose RNN with LSTM cells for "remembering" cell state over arbitrary time tervals

orget gate - what to forget:

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

put gate - what to save:

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

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

odate cell state:

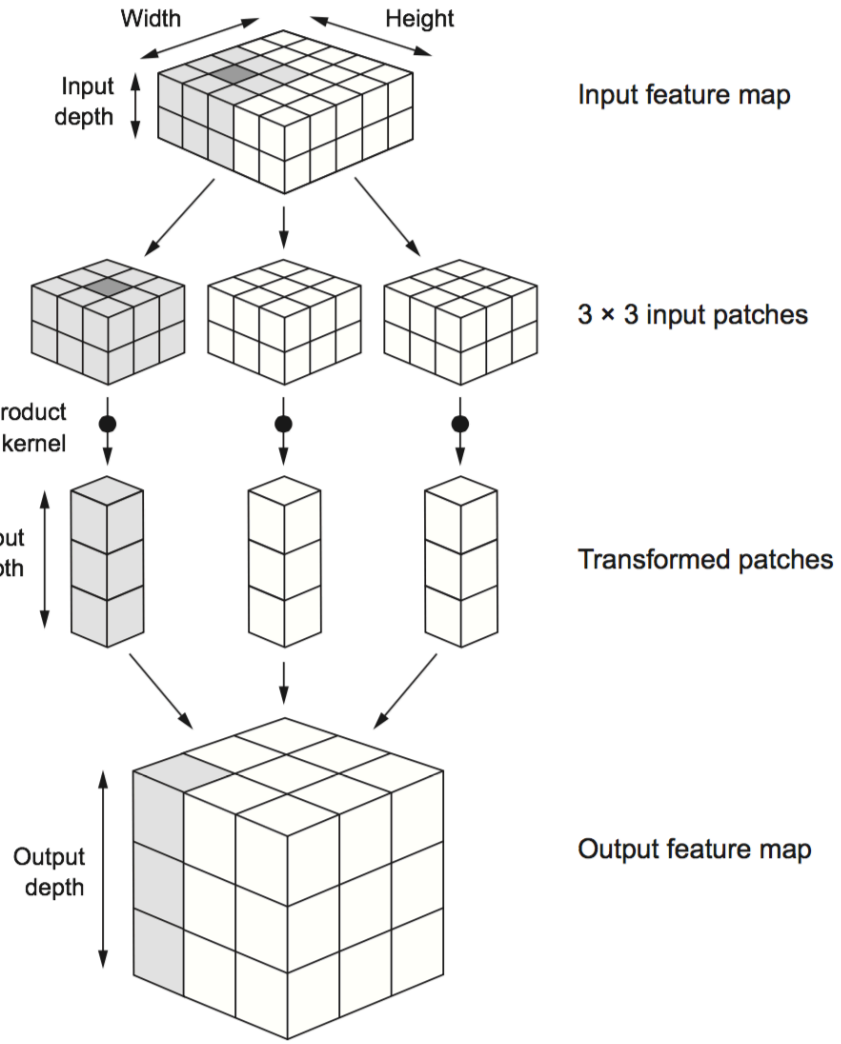
$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

utput:

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

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

Convolution Neural Network (ConvNet)

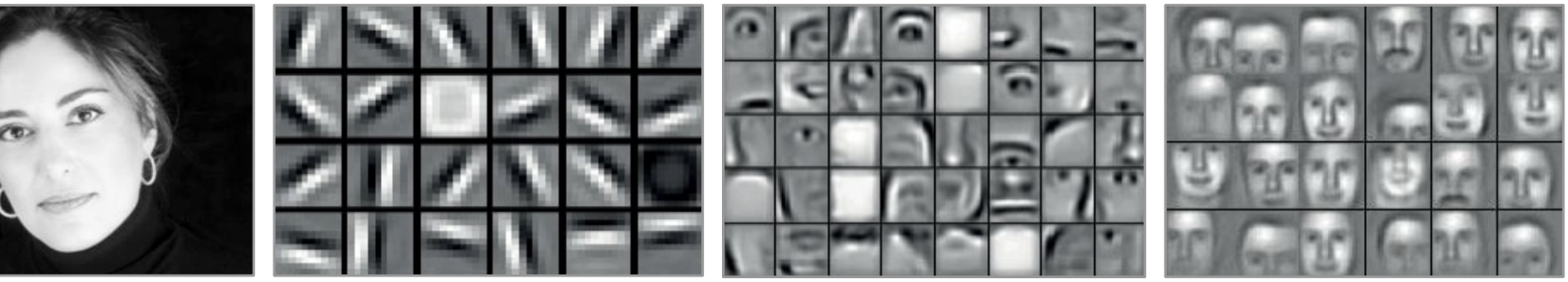


- A ConvNet is made up of Layers: convolut pooling, fully connected.
- Each layer transforms an input 3D volume an output 3D volume.
- Learns local parameters in input space us stack of (e.g., 3x3) convolution filters
- The patterns the filters learn are translat invariant.
- Learn spatial hierarchies of patterns by a conv and pooling layers. E.g., edges, sha motifs.

FRANÇOIS CHOLLET, 2018

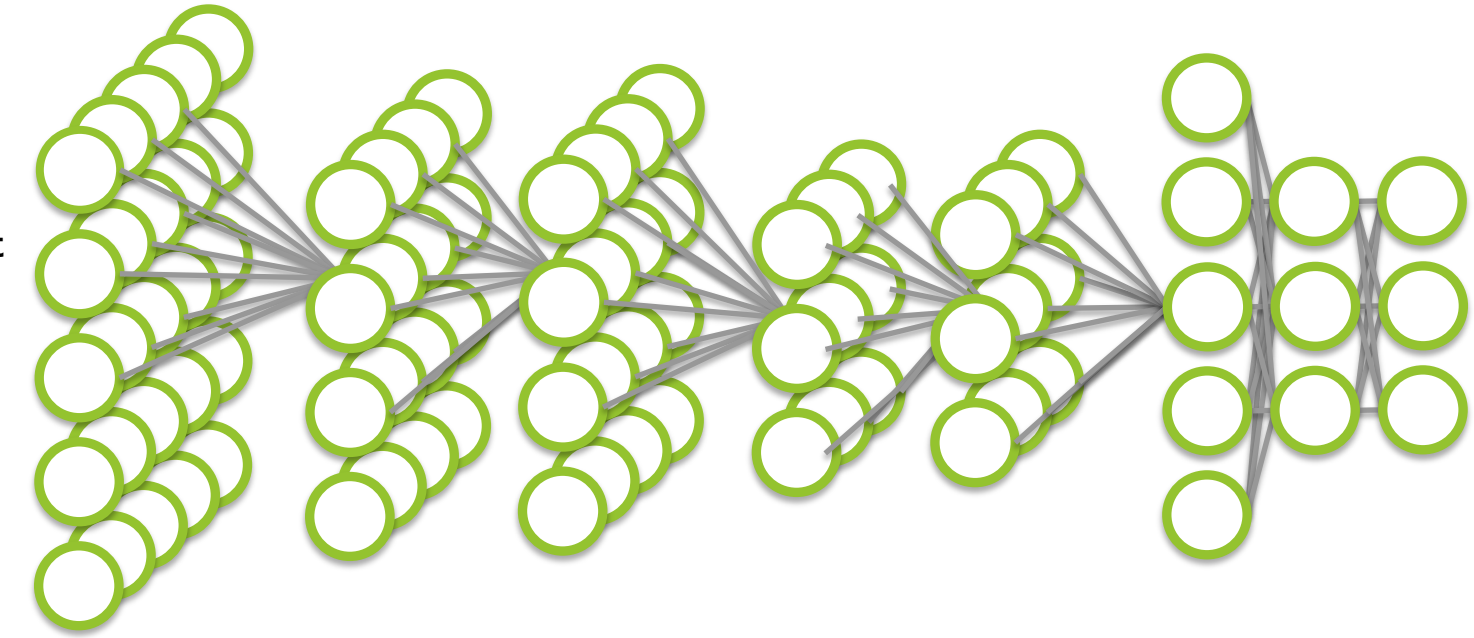
DEEP LEARNING APPROACH - TRAINING

aw data | Low-level features | Mid-level features | High-level features



Forward propagation

Backward propagation



Process

- Forward propagati yields an inferred l for each training in
- Loss function used calculate differenc between known lab and predicted labe each image
- Weights are adjust during backward propagation
- Repeat the process

Experiment #1 - WISDM Activity Dataset

mporal sensor analysis and activity tracking

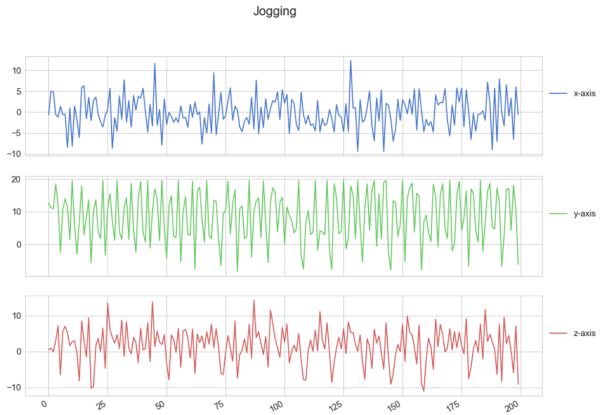
user	activity	timestamp	x-axis	y-axis	z-axis
33	Jogging	49105962326000	-0.694638	12.680544	0.503953
33	Jogging	49106062271000	5.012288	11.264028	0.953424
33	Jogging	49106112167000	4.903325	10.882658	-0.081722
33	Jogging	49106222305000	-0.612916	18.496431	3.023717
33	Jogging	49106332290000	-1.184970	12.108489	7.205164

Layer (type)	Output Shape	Para
conv1d_9 (Conv1D)	(None, 100, 64)	1024
max_pooling1d_9 (MaxPooling1	(None, 50, 64)	0
dropout_9 (Dropout)	(None, 50, 64)	0
conv1d_10 (Conv1D)	(None, 50, 128)	4108
max_pooling1d_10 (MaxPooling	(None, 25, 128)	0
dropout_10 (Dropout)	(None, 25, 128)	0
gru_5 (GRU)	(None, 25, 50)	2685
flatten_5 (Flatten)	(None, 1250)	0
dense_8 (Dense)	(None, 6)	7506

Total params: 76,468

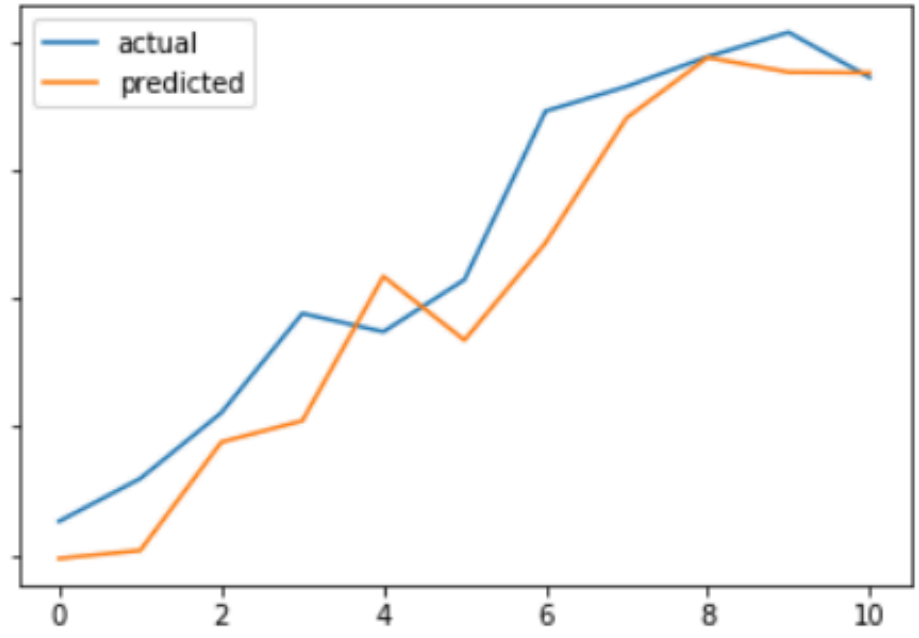
Test accuracy: 0.985

	Accuracy
r LSTM	94.2
r LSTM	95.6
r Convnet	94.5
r Convnet	98.7
r Convnet + 1-layer LSTM	98.5



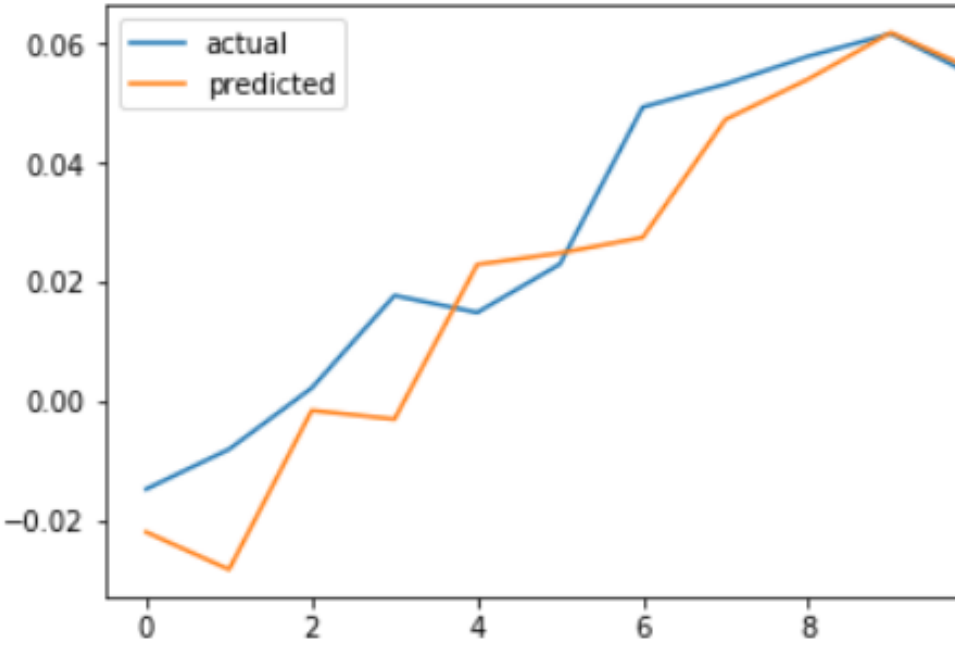
xperiment #2 - Sequence Prediction - Stoc Prices

ayer LSTM



Mean Absolute Error is: 0.00810866830379987

Convnet + 2 layer LSTM



The Mean Absolute Error is: 0.0086144270502

Experiment #3 – Multi-label classification

dict stack overflow posting labels – 2 – layer biLSTM, embeddings, softmax

---------- Train set quality: ----------
cessed 105778 tokens with 4489 phrases; found: 4532 phrases; correct: 4389.
cision: 96.84%; recall: 97.77%; F1: 97.31

---------- Test set quality: ----------
cessed 13258 tokens with 604 phrases; found: 479 phrases; correct: 229.
cision: 47.81%; recall: 37.91%; F1: 42.29

	title	
0	How to draw a stacked dotplot in R?	
1	mysql select all records where a datetime fiel...	
2	How to terminate windows phone 8.1 app	
3	get current time in a specific country via jquery	[jav
4	Configuring Tomcat to Use SSL	
5	Awesome nested set plugin - how to add new chi...	
6	How to create map from JSON response in Ruby o...	[ruby, ruby-c
7	rspec test if method is called	
8	SpringBoot Catalina LifeCycle Exception	[java, sprin
9	How to import data from excel to mysql databas...	[ph

Takeaways

tures
Episodic temporal data
Relational models
Machine reading/summarization
One-shot learning
Online learning
Model interpretability

sics - Know your data
LSTMs are sloooow
Never start with the most complex model, establish a baseline with a basic mod
Deep learning works well when you have very large labeled datasets
DL lacks interpretability
For many tasks, XGBoost/Boosted Trees, SVM, and statistical models work well

References

ns Wei Yu, David Dohan, Minh-Thang Luong et al., QANET: COMBINING LOCAL CONVOLUTION WITH
BAL SELF-ATTENTION FOR READING COMPREHENSION
s://arxiv.org/pdf/1804.09541.pdf

av Rajpurkar*, Awni Hannun*, Masoumeh Haghpanahi, Codie Bourn, and Andrew Ng, Cardiologist-Level Arrhythmia Detection With Convolutional Ne
orks
s://stanfordmlgroup.github.io/projects/ecg/

oodfellow, Yoshua Bengio and Aaron Courville, Deep Learning
//www.deeplearningbook.org/

Olah, Understanding LSTM Networks
//colah.github.io/posts/2015-08-Understanding-LSTMs/

Olah, Conv Nets a Modular Perspective
//colah.github.io/posts/2014-07-Conv-Nets-Modular/

ej karpathy, The Unreasonable Effectiveness of Recurrent Neural Network
//karpathy.github.io/2015/05/21/rnn-effectiveness/

ej Karpatyh, Convolutional Neural Networks for Visual Recognition
//karpathy.github.io/2015/05/21/rnn-effectiveness/

çois Chollet, Deep Learning with Python
s://www.manning.com/books/deep-learning-with-python

lien Géron, Hands-On Machine Learning with Scikit-Learn and TensorFlow
epts, Tools, and Techniques to Build Intelligent Systems
//shop.oreilly.com/product/0636920052289.do