Deep Learning for Bio-medical Applications

Vinayakumar R

PhD Student,

Centre for Computational Engineering and Networking, Amrita Vishwa Vidyapeetham,

Coimbatore

https://vinayakumarr.github.io/

Agenda

AMRITA
VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY

- Why deep learning?
- Deep Neural Networks
- Recurrent structures – RNN, LSTM, GRU
- Bidirectional recurrent structures
- Hands on tutorial on python – numpy, scipy, matplotlib, pandas
- Hands on tutorial on TensorFlow and Keras
- Bio-medical usecases – Sleep Apnea Detection, ECG Analysis – Atrial fibrillation, Sleep Apnea, Anomaly detection in Phonocardiogram

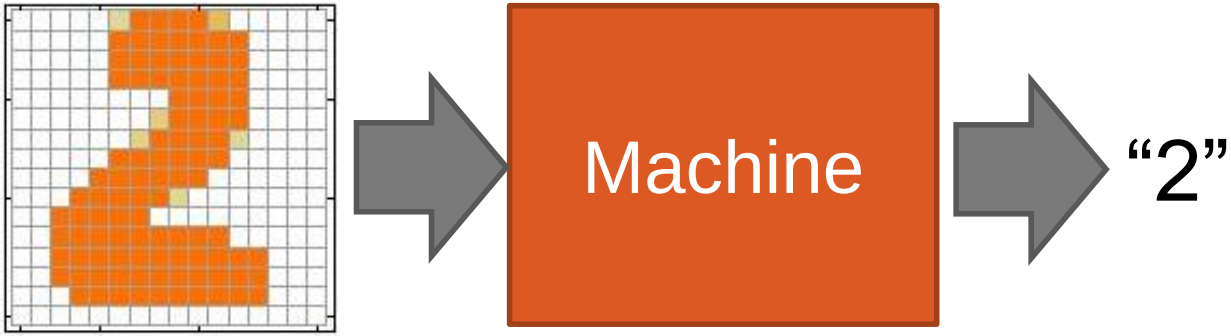
Deep Learning

AMRITA
VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY

- Deep learning is kind of hard. Why bother with it?
- Amazing results… in speech, NLP, vision/multimodal work
- Does its own feature selection!
- The big players (Google, Facebook, Baidu, Microsoft, IBM…) are doing a lot of this
- The hot new thing?
- Actually, many of the architectures that we'll talk about were invented in the 1980s and 1990s
- What's new is hardware that can use these architectures at scale.

Example Application

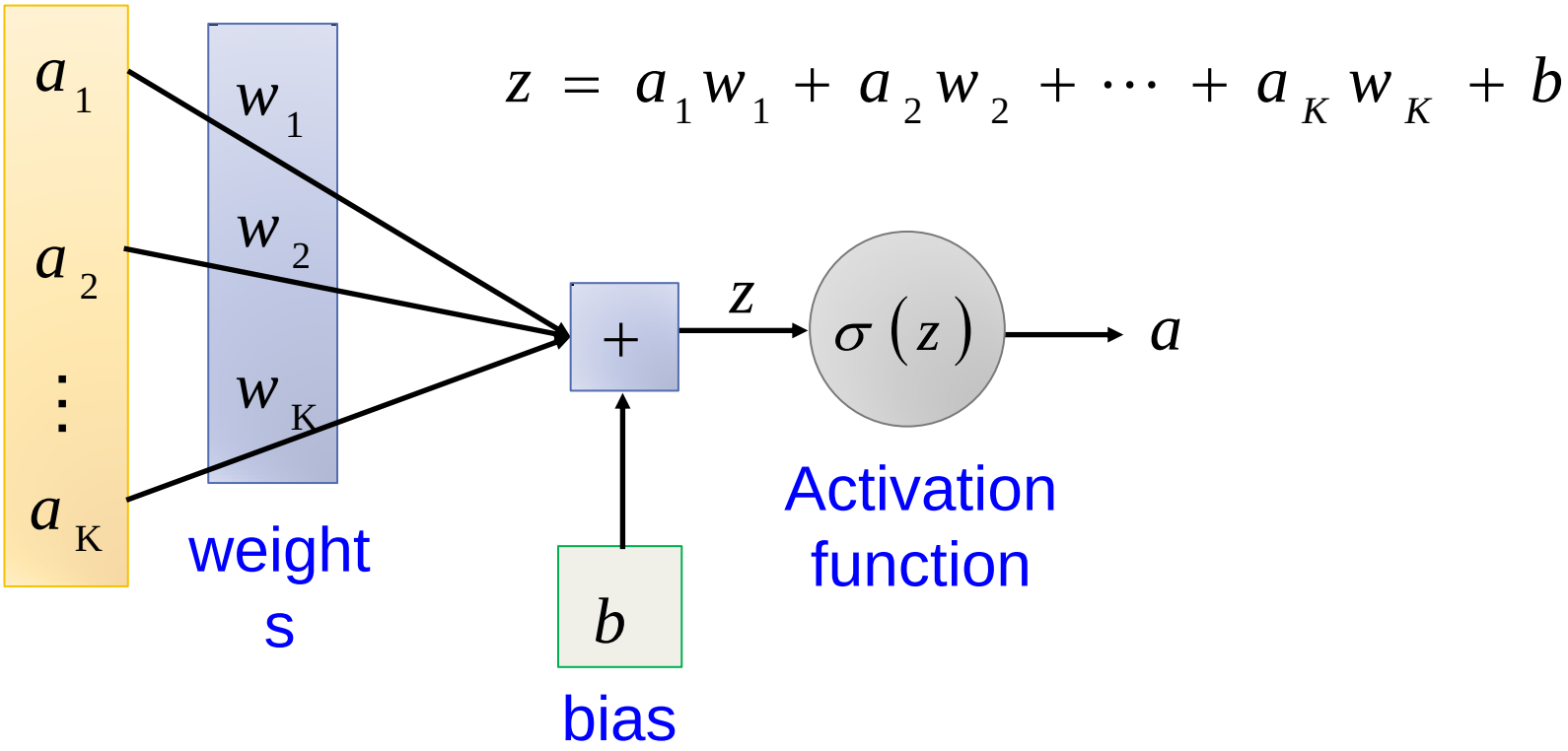
- Handwriting Digit Recognition



AMRITA VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY

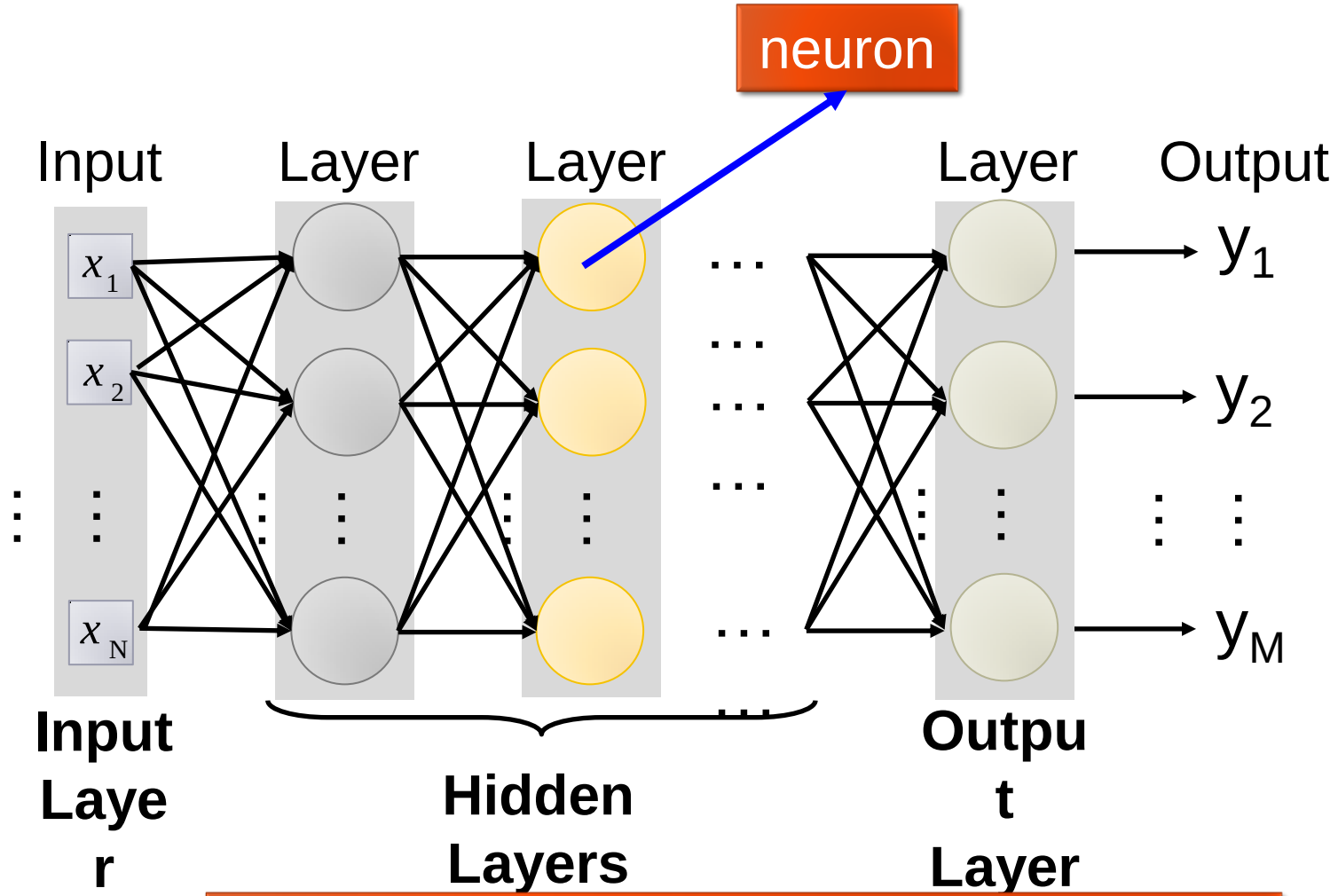
Element of Neural Network

Neuron $f: R^K \rightarrow R$



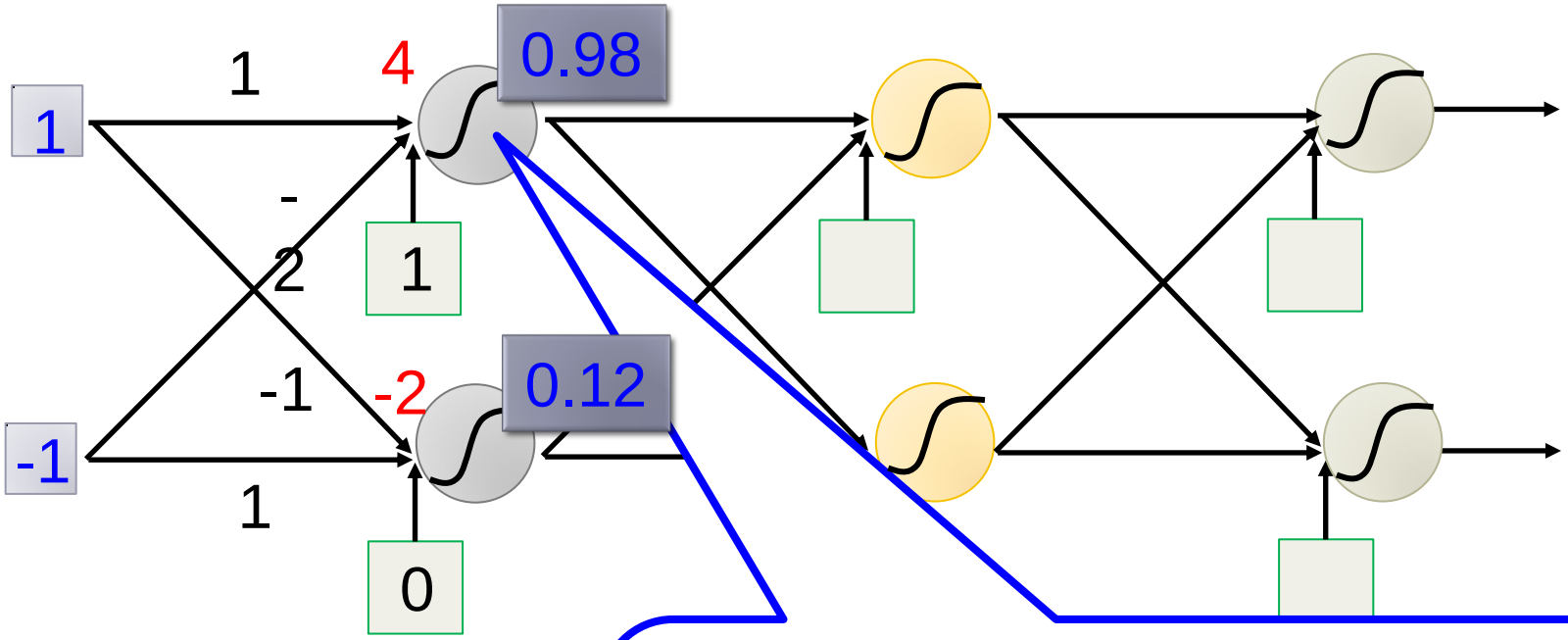
$$z = a_1 w_1 + a_2 w_2 + \cdots + a_K w_K + b$$

Neural Network



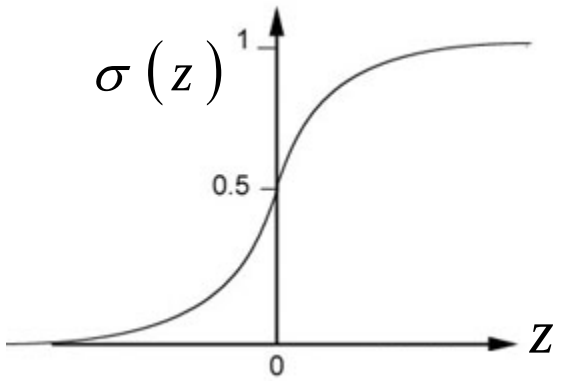
Deep means many hidden layers

Neural Network

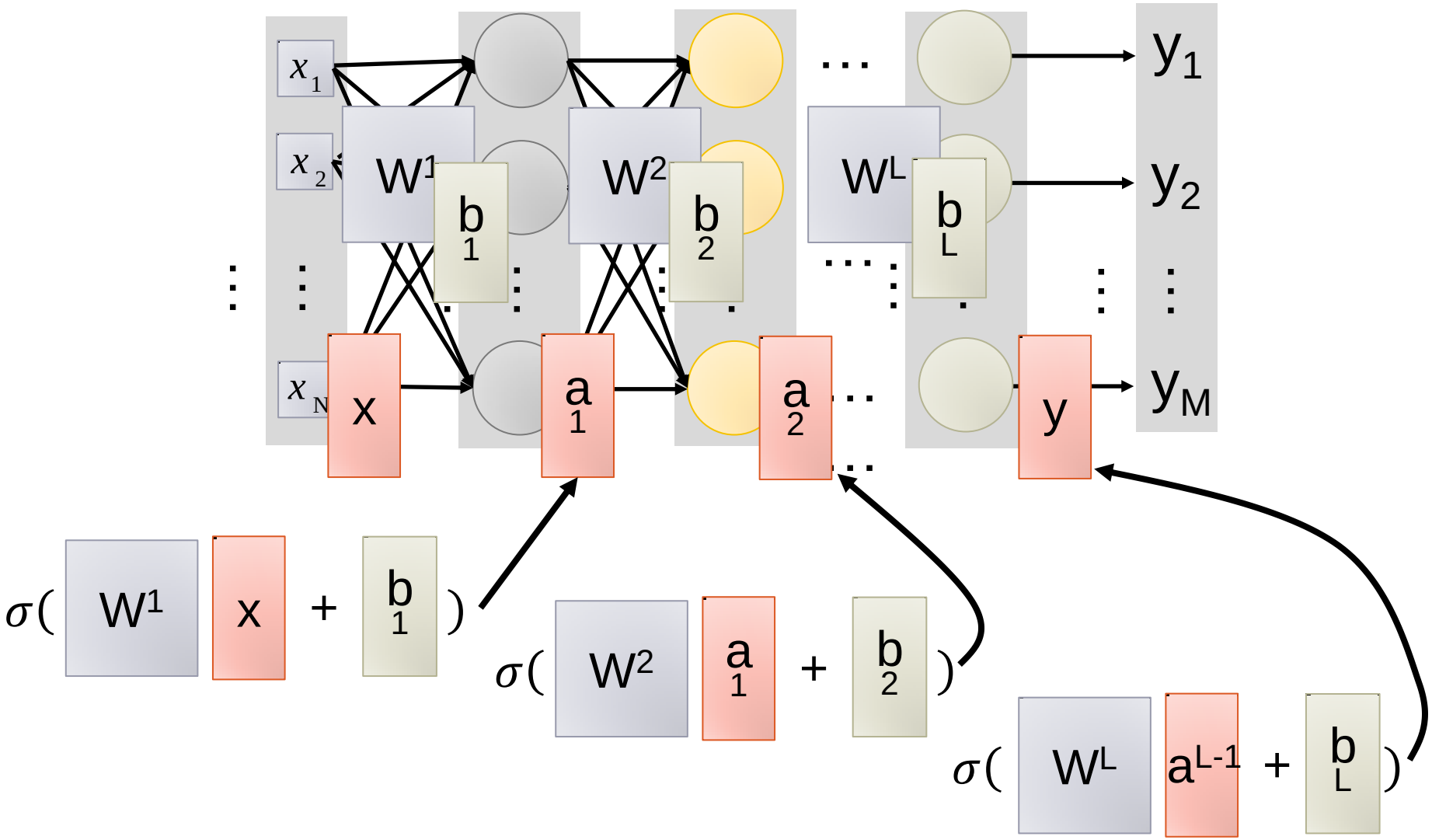


Sigmoid Function

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



Neural Network

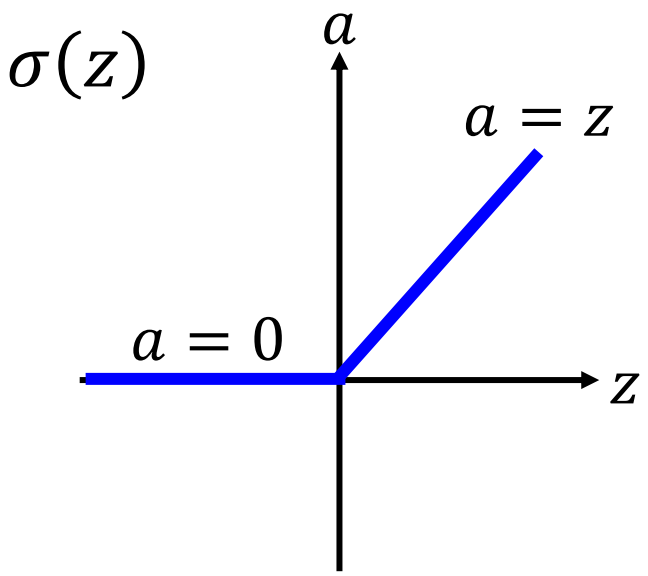


Training DNN

New Activation Function

ReLU

- Rectified Linear Unit (ReLU)



Reason:

1. Fast to compute

2. Vanishing gradient problem

$$f'(x) = \begin{cases} 1 \; if \; x >= 0 \\ 0 \; if \; x < 0 \end{cases}$$

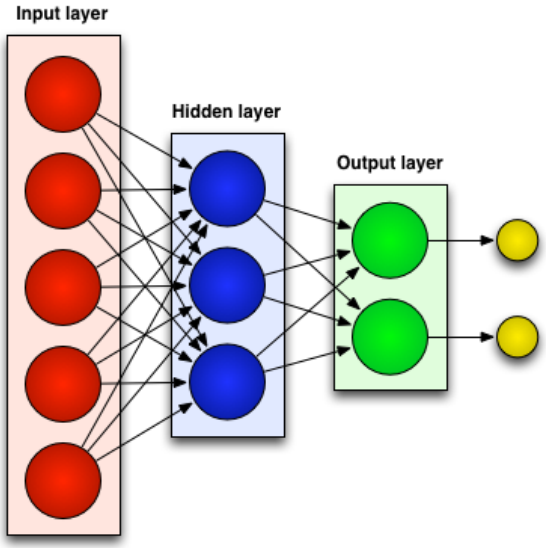
$$\sigma'(z) = \sigma(z) * (1 - \sigma(z))$$

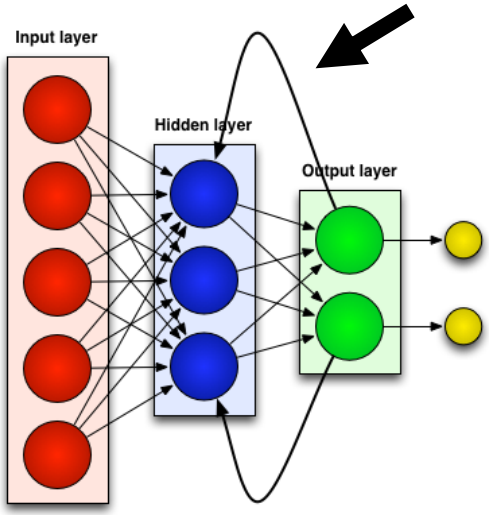
Neural Networks

- **Generally there are two kinds of neural networks:**

➢ **Feedforward Neural Networks:**

✓ **connections between the units do not form a cycle**



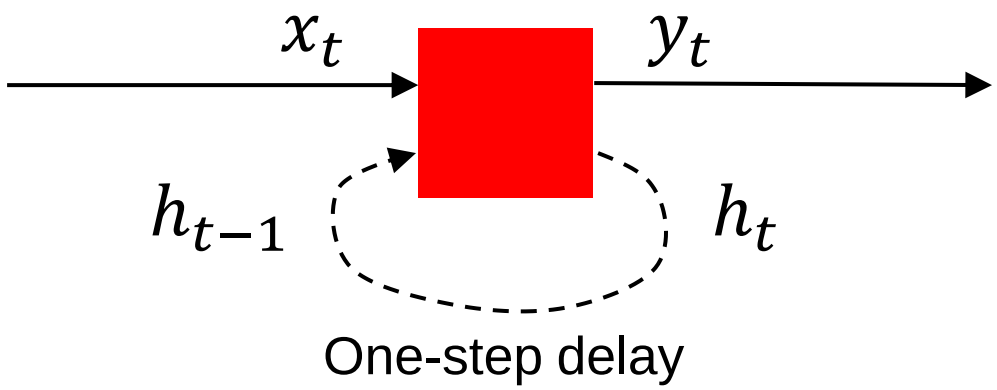


➢ **Recurrent Neural Network:**

✓ **connections between units form cyclic paths**

Recurrent Neural Networks

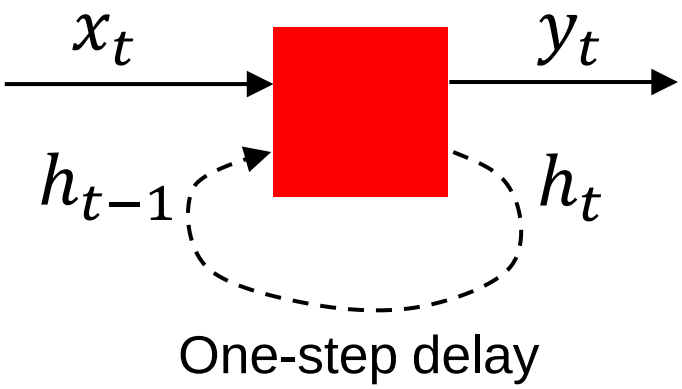
Recurrent networks introduce cycles and a notion of time.

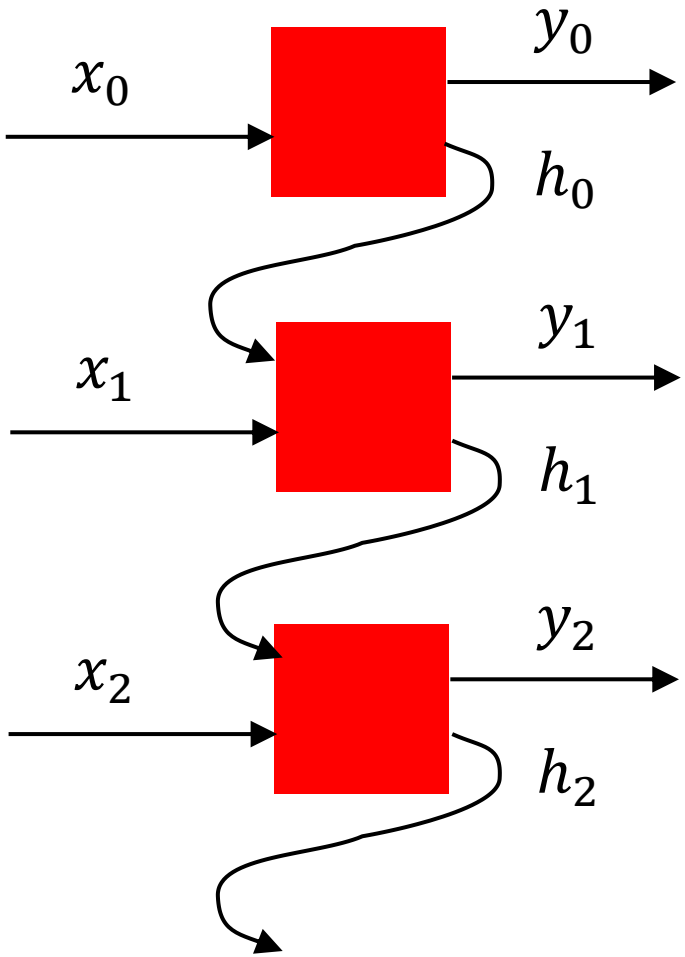


- They are designed to process sequences of data $x_1, \ldots, x_n$ and can produce sequences of outputs $y_1, \ldots, y_m$.

Unrolling RNNs

RNNs can be unrolled across multiple time steps.



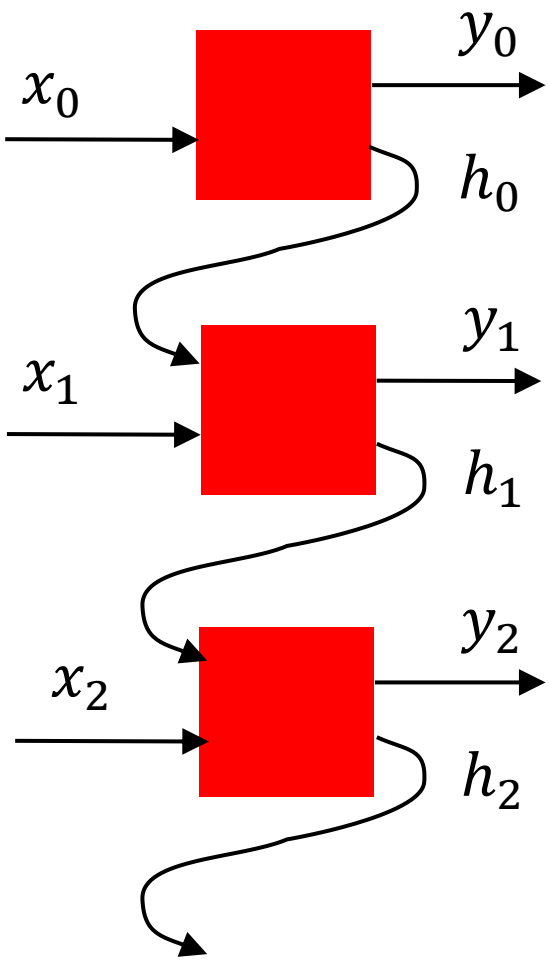


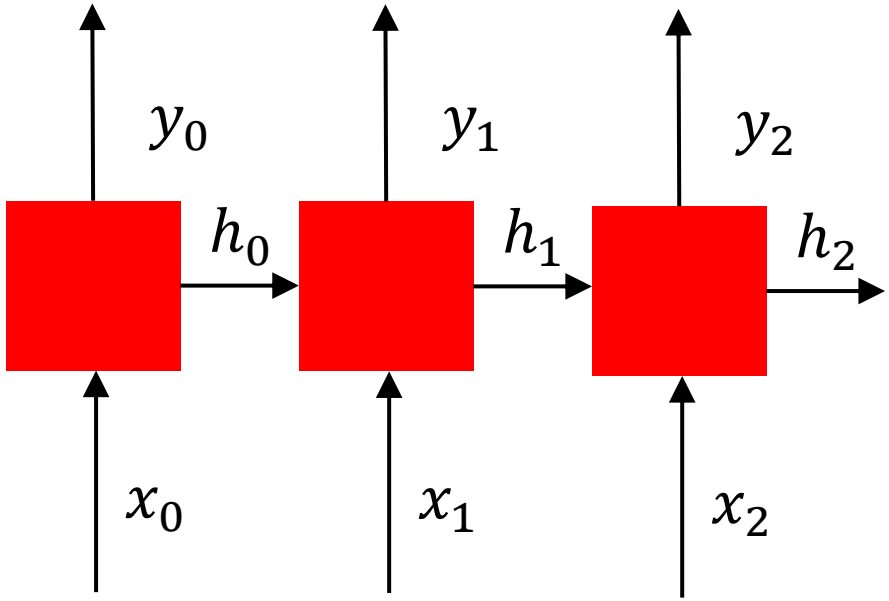
This produces a DAG which supports backpropagation.

But its size depends on the input sequence length.

Unrolling RNNs

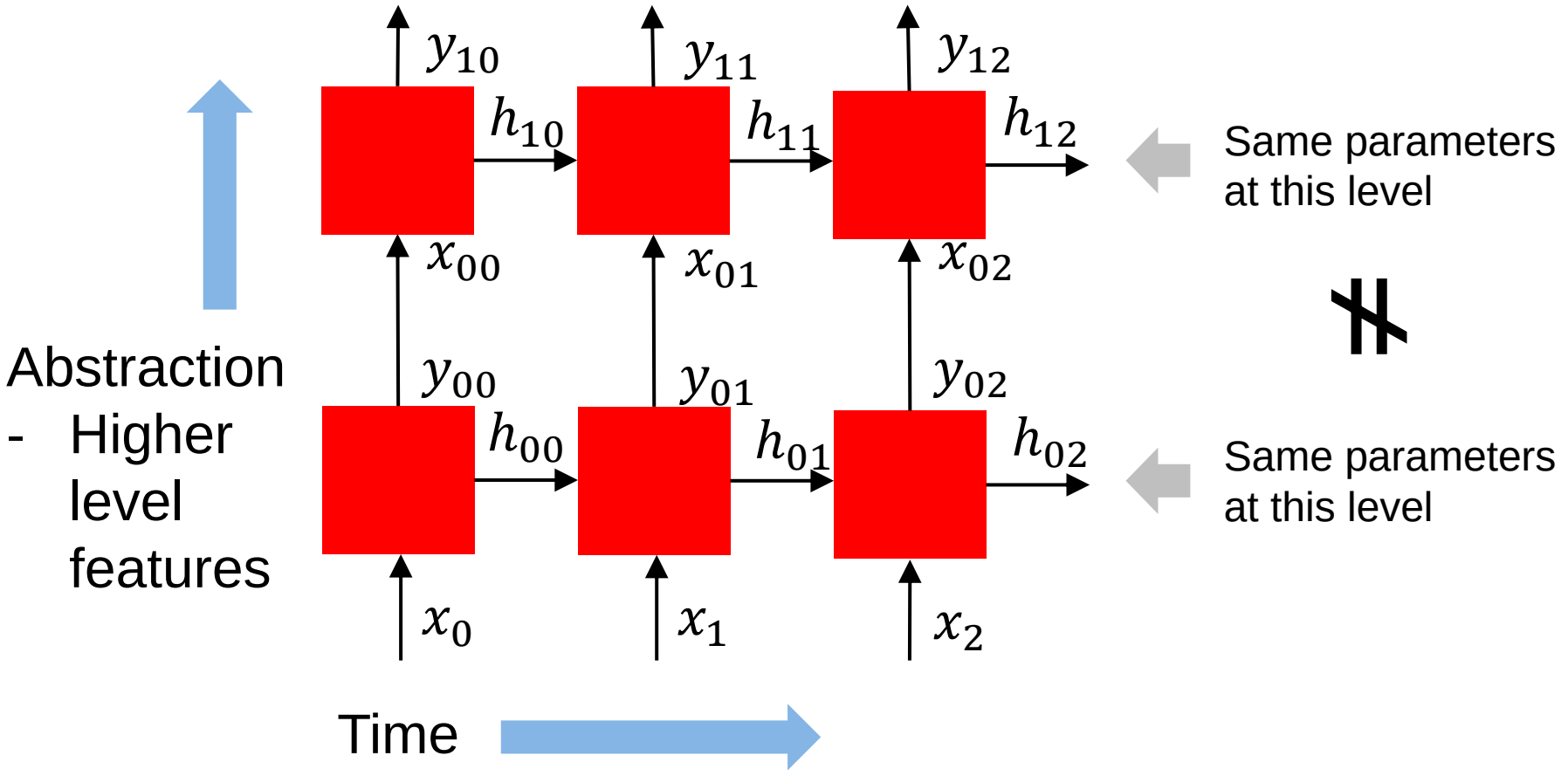
Usually drawn as:





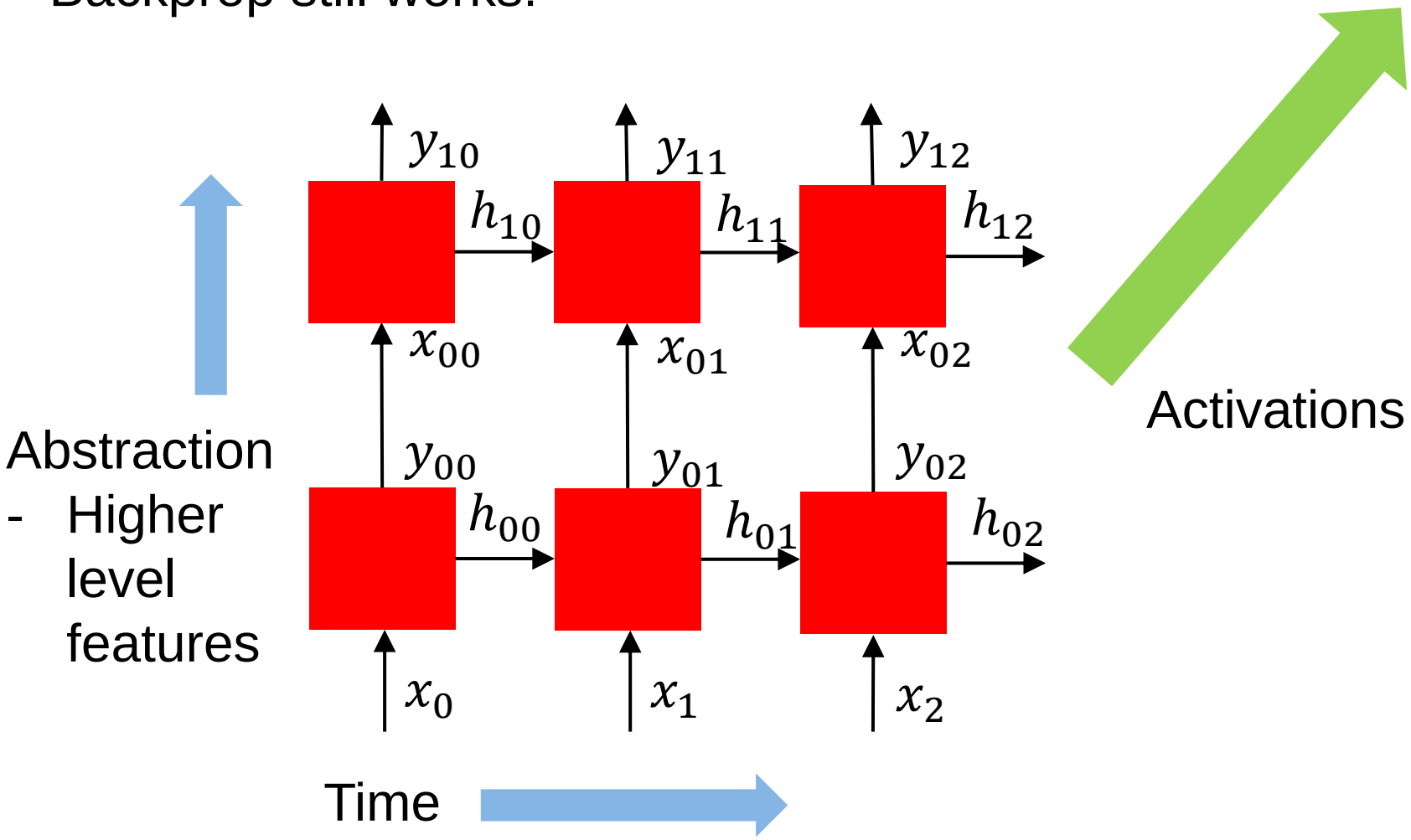
RNN structure

Often layers are stacked vertically (deep RNNs):



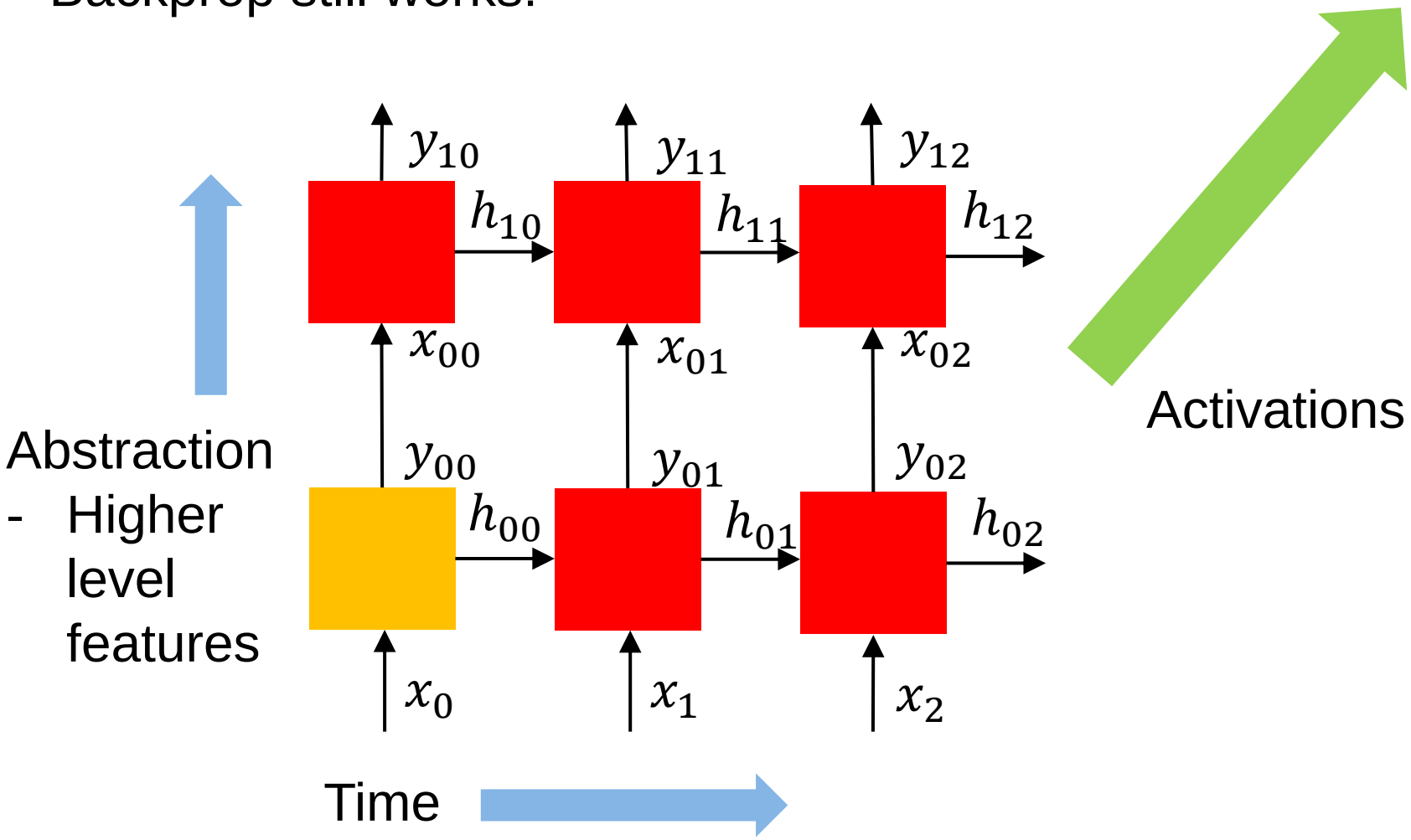
RNN structure

Backprop still works:



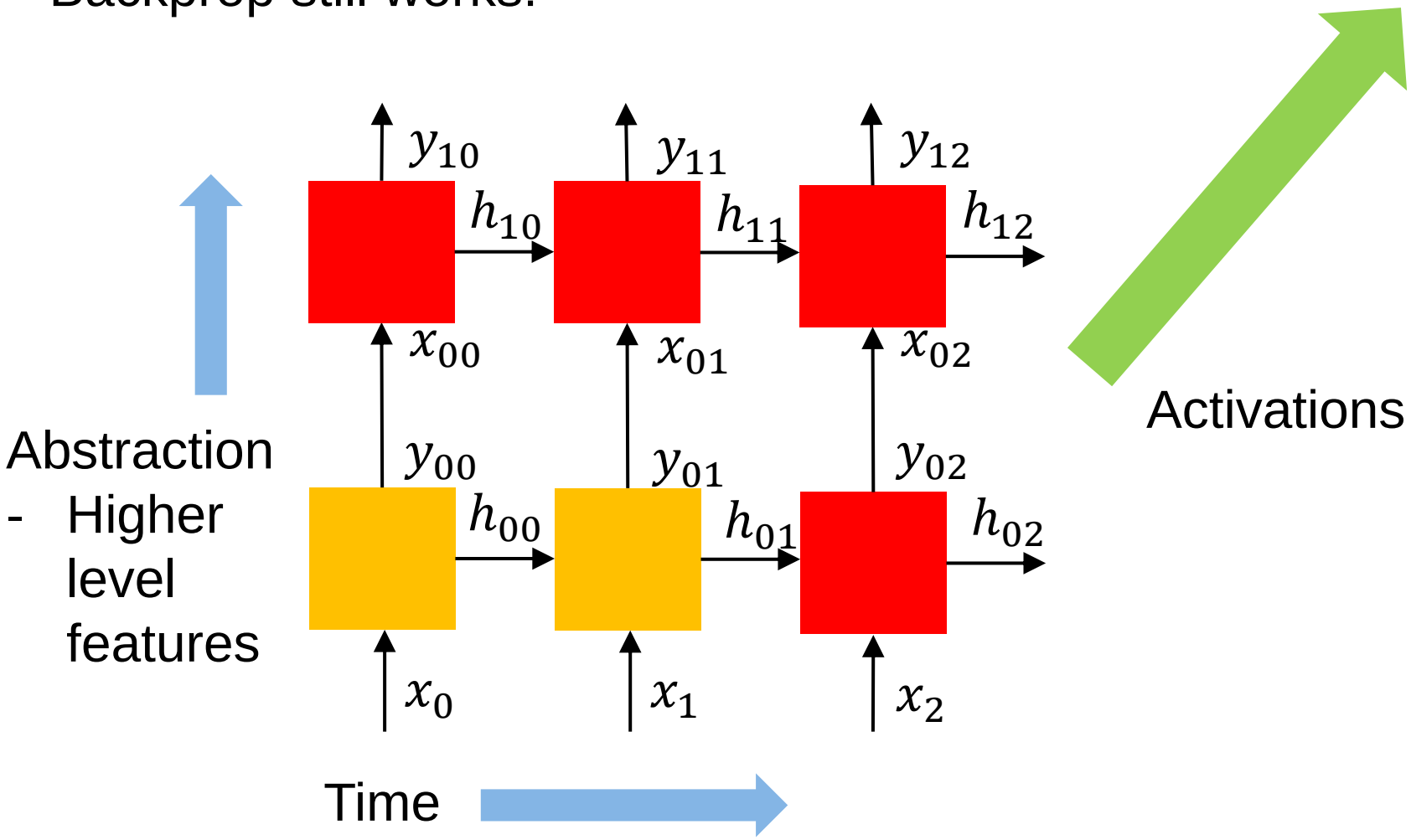
RNN structure

Backprop still works:



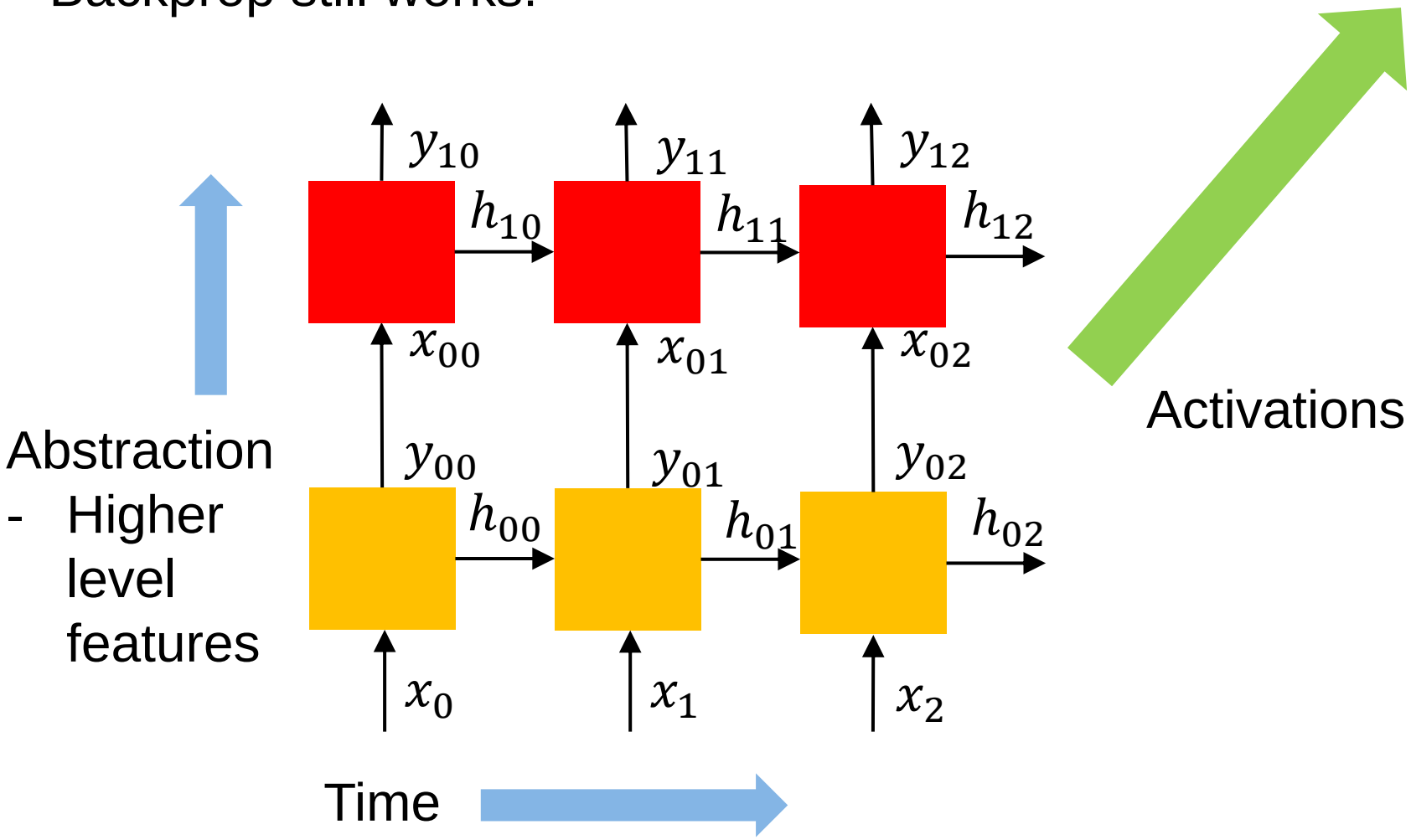
RNN structure

Backprop still works:



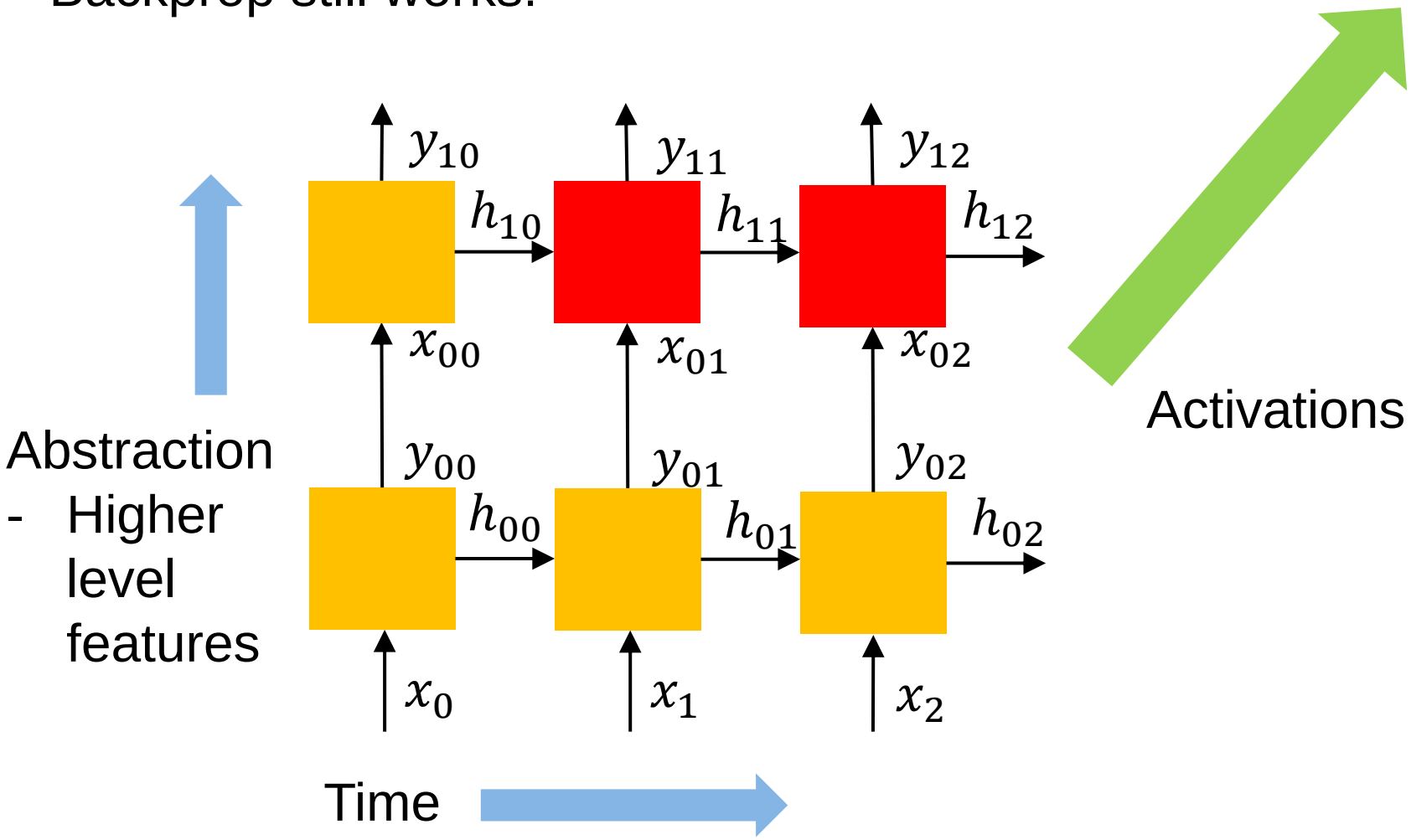
RNN structure

Backprop still works:



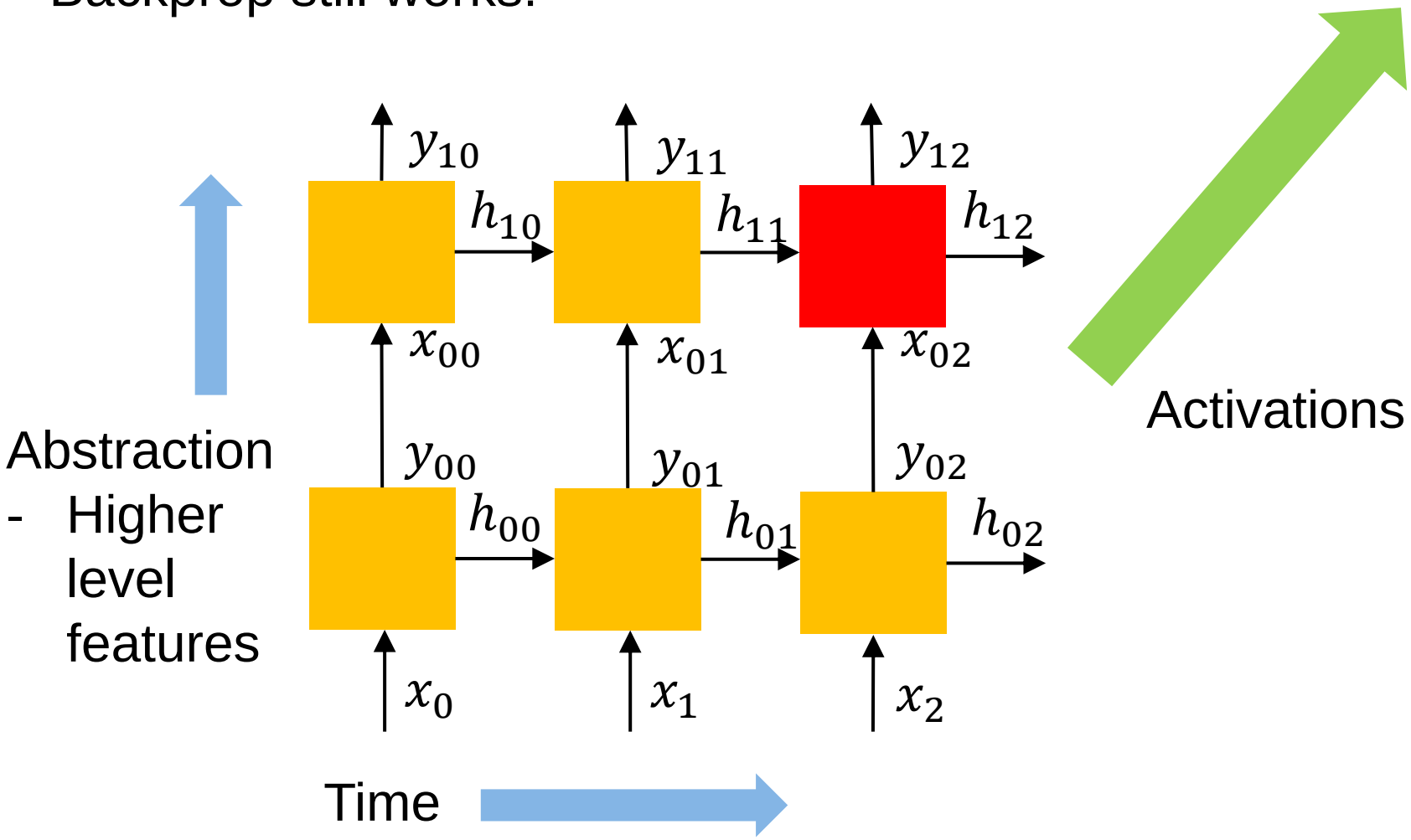
RNN structure

Backprop still works:



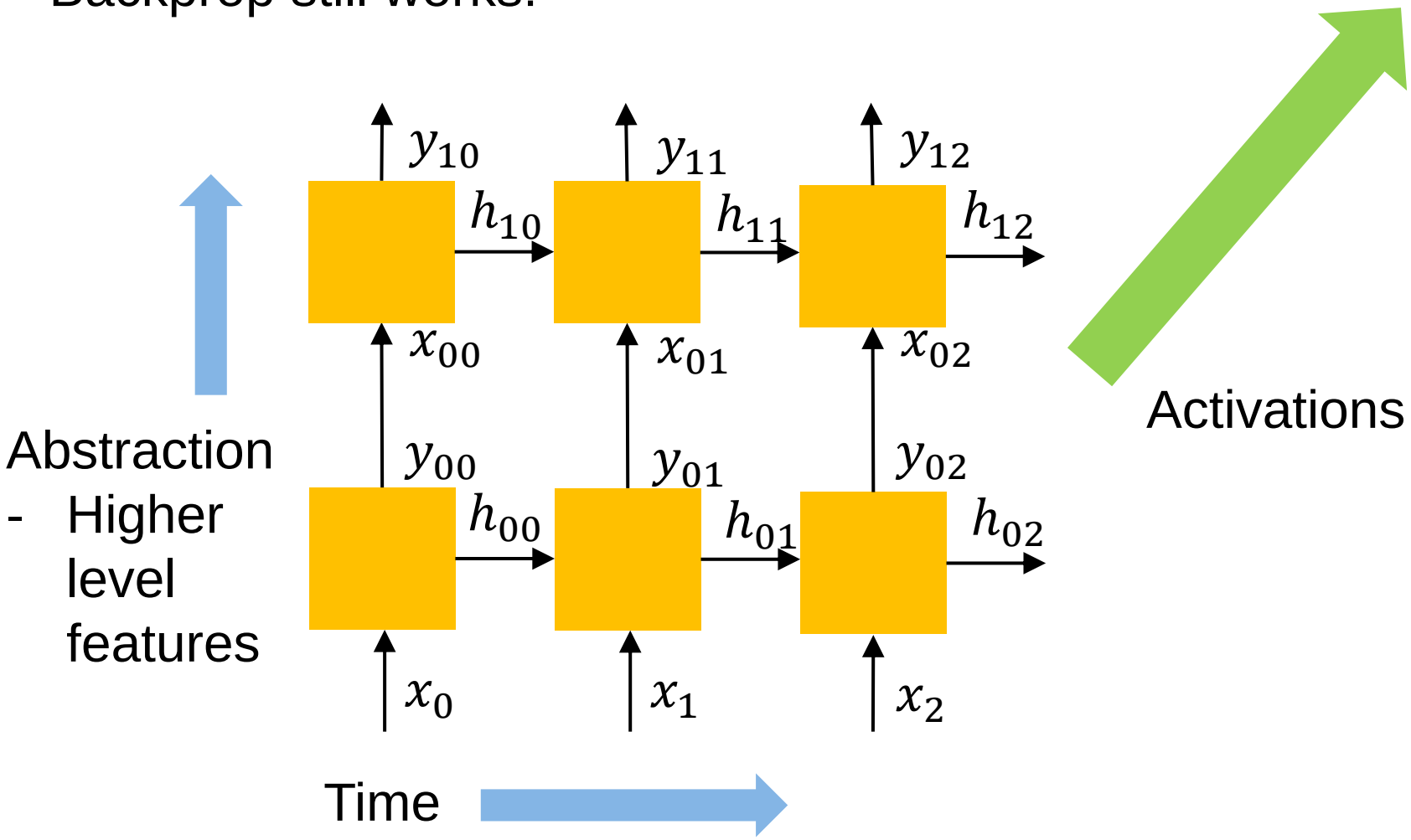
RNN structure

Backprop still works:



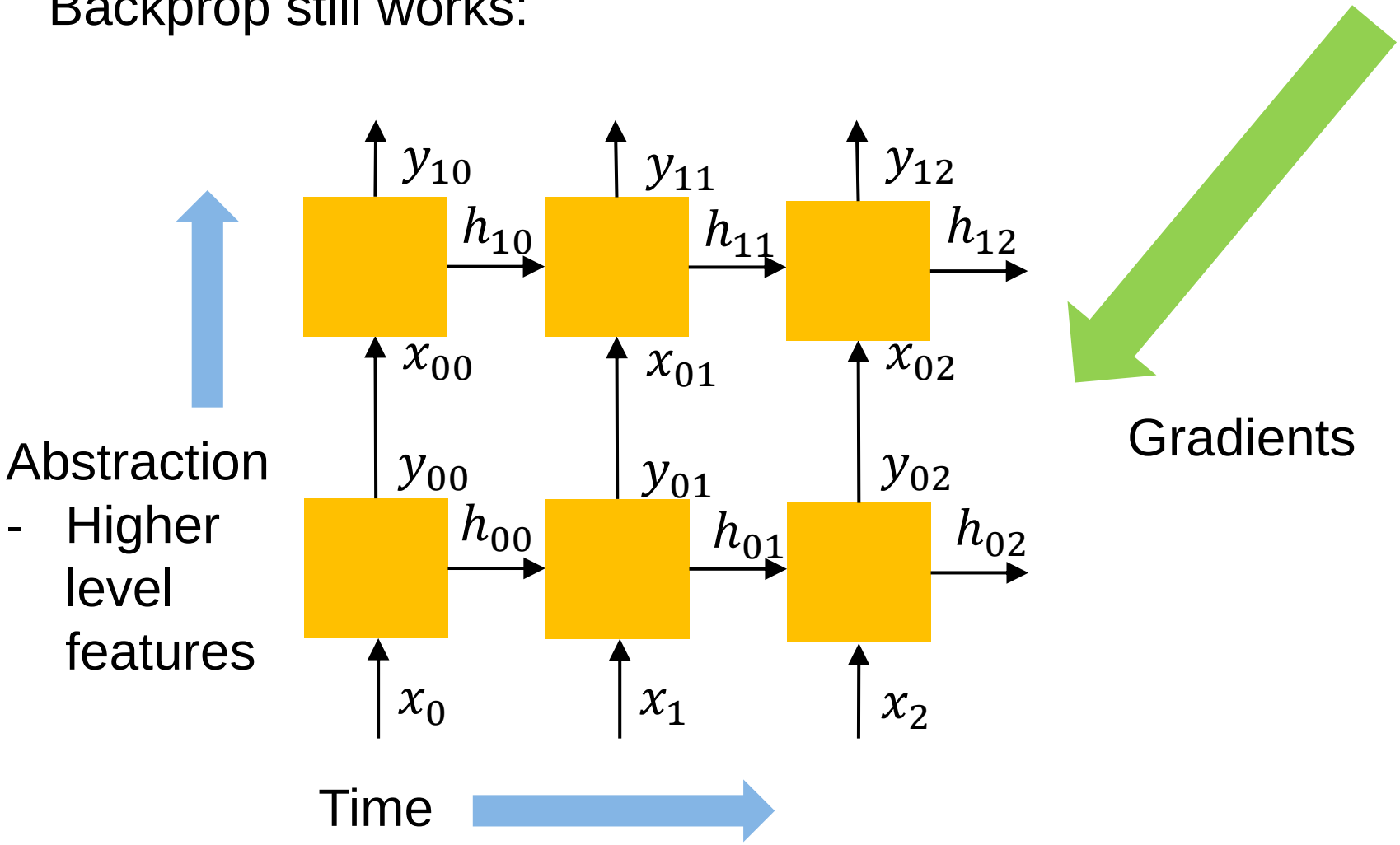
RNN structure

Backprop still works:



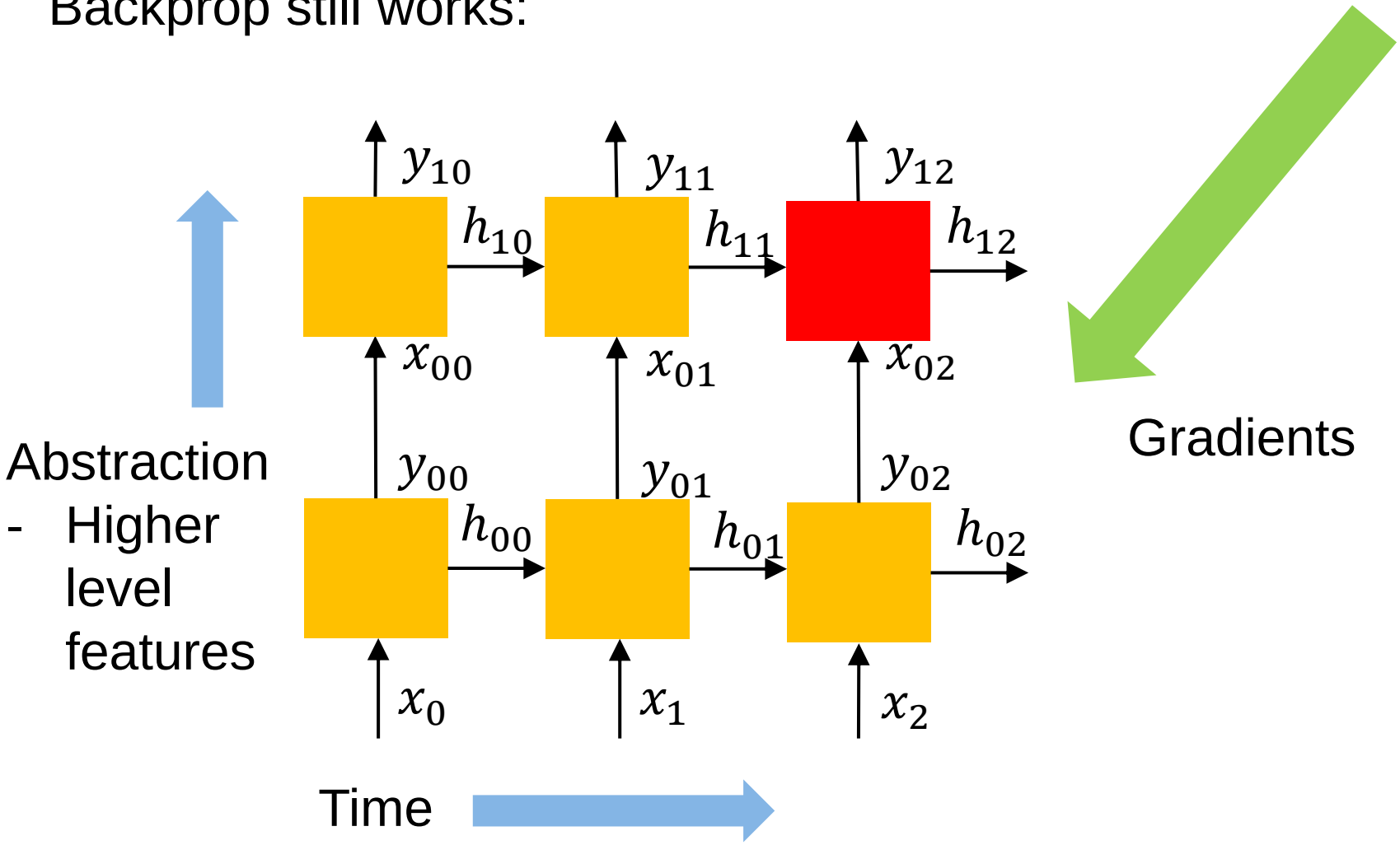
RNN structure

Backprop still works:



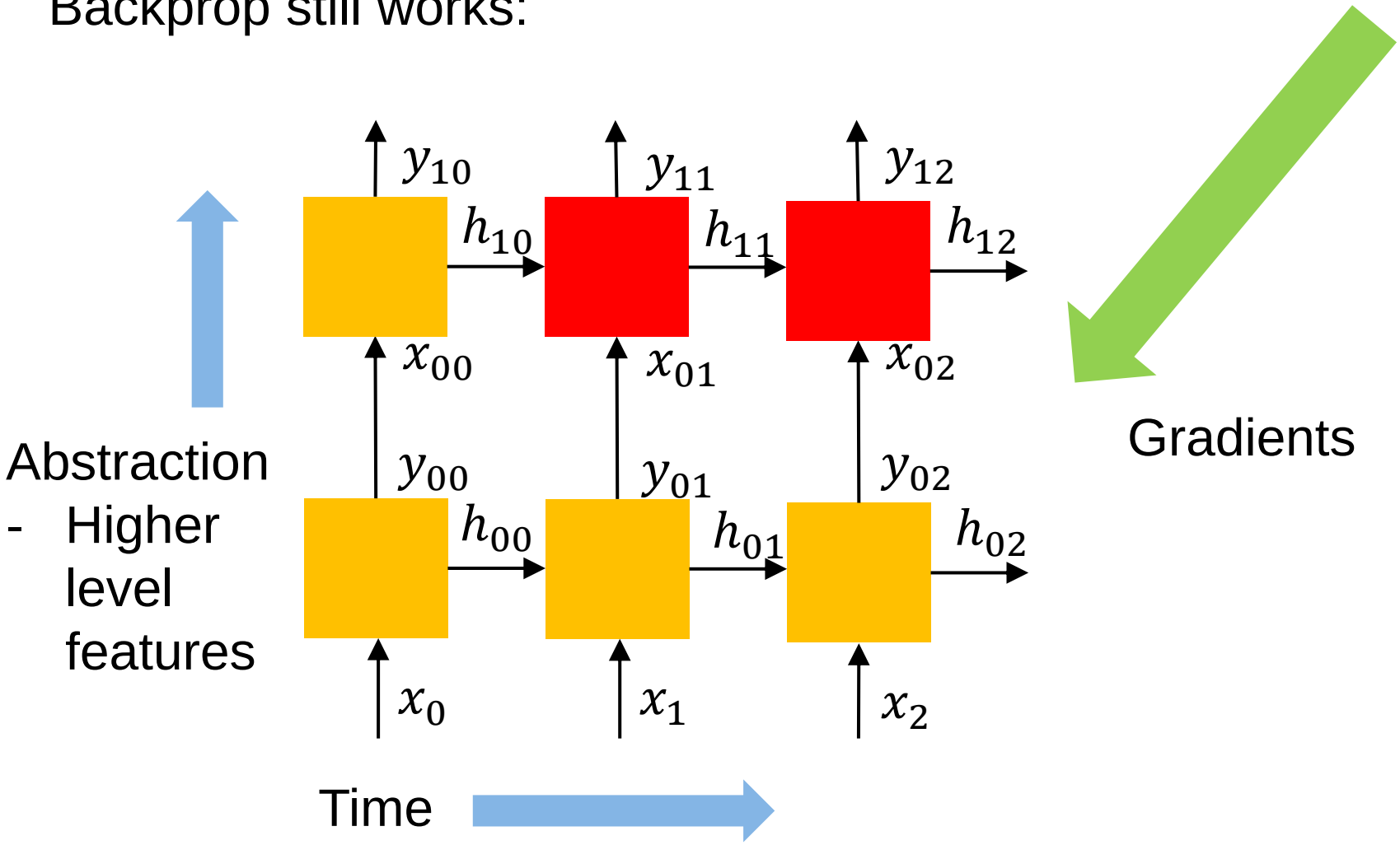
RNN structure

Backprop still works:



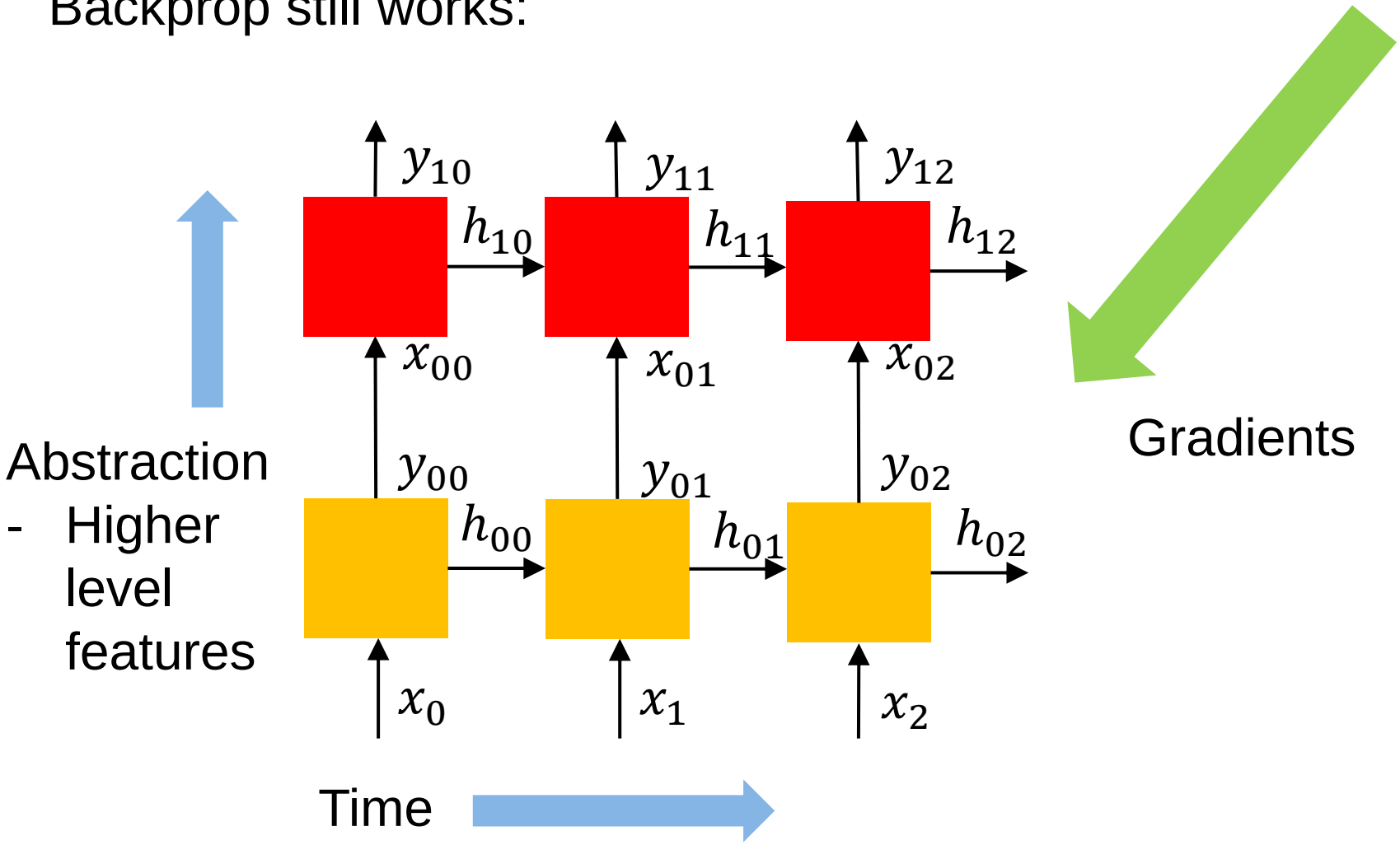
RNN structure

Backprop still works:



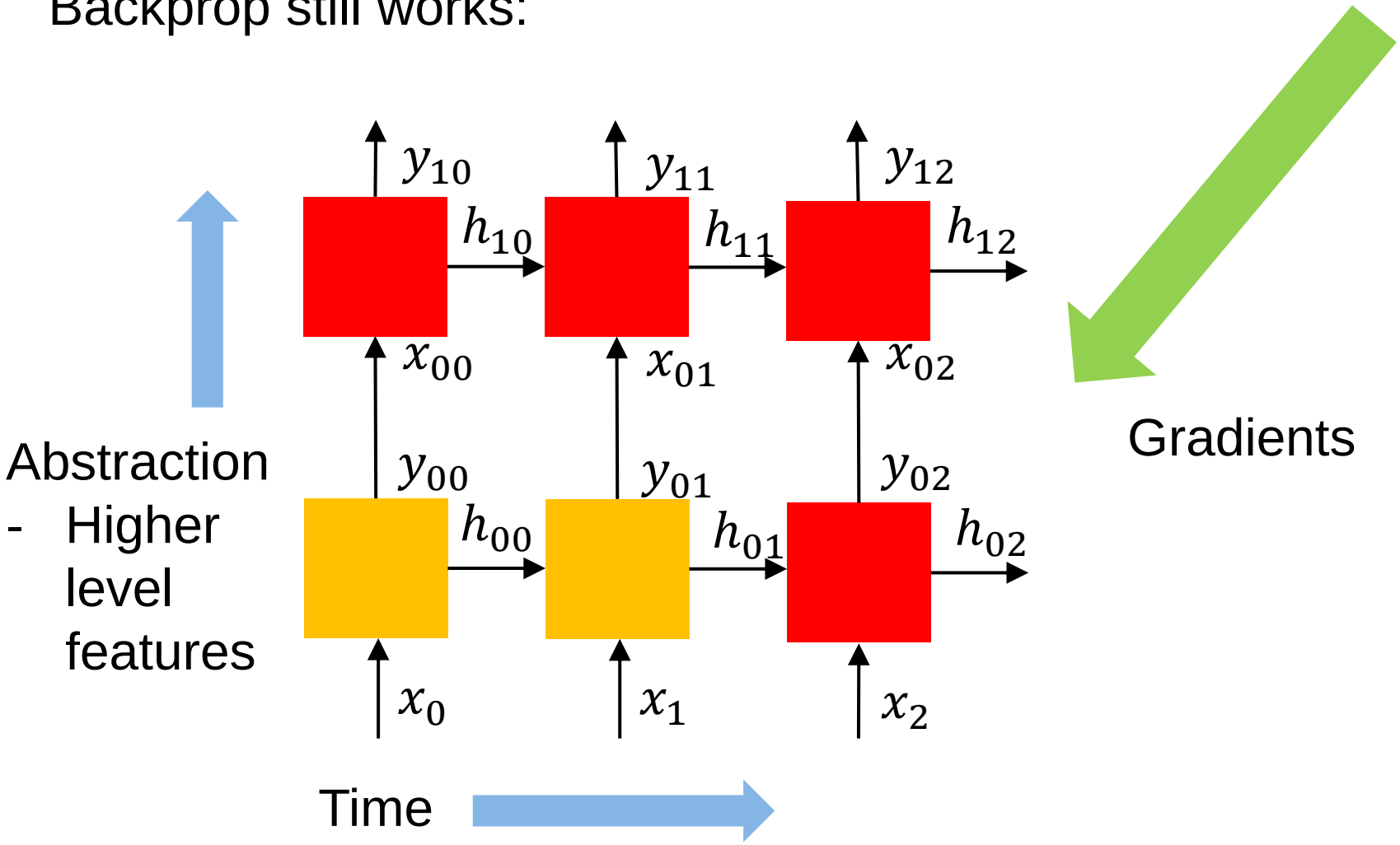
RNN structure

Backprop still works:



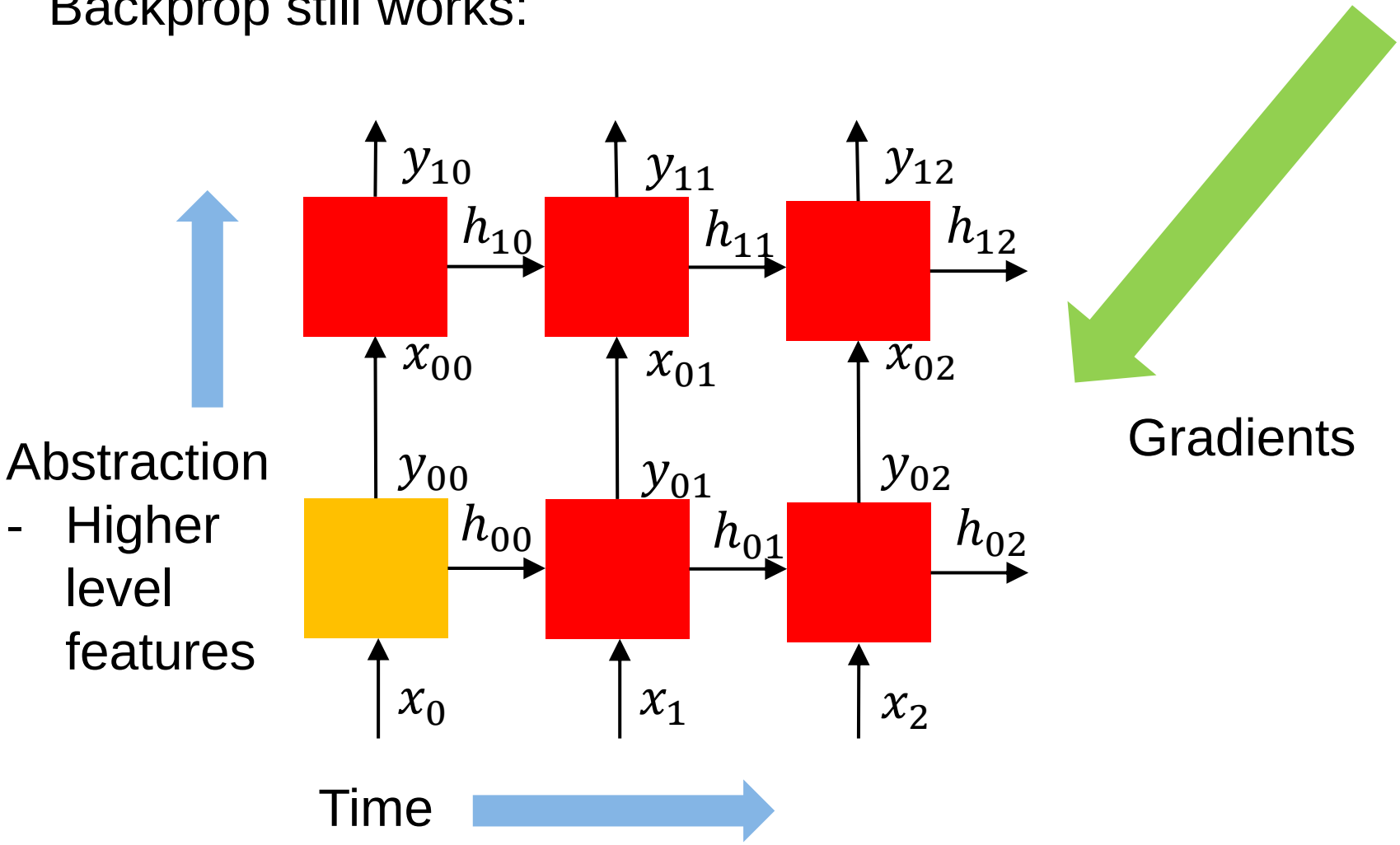
RNN structure

Backprop still works:



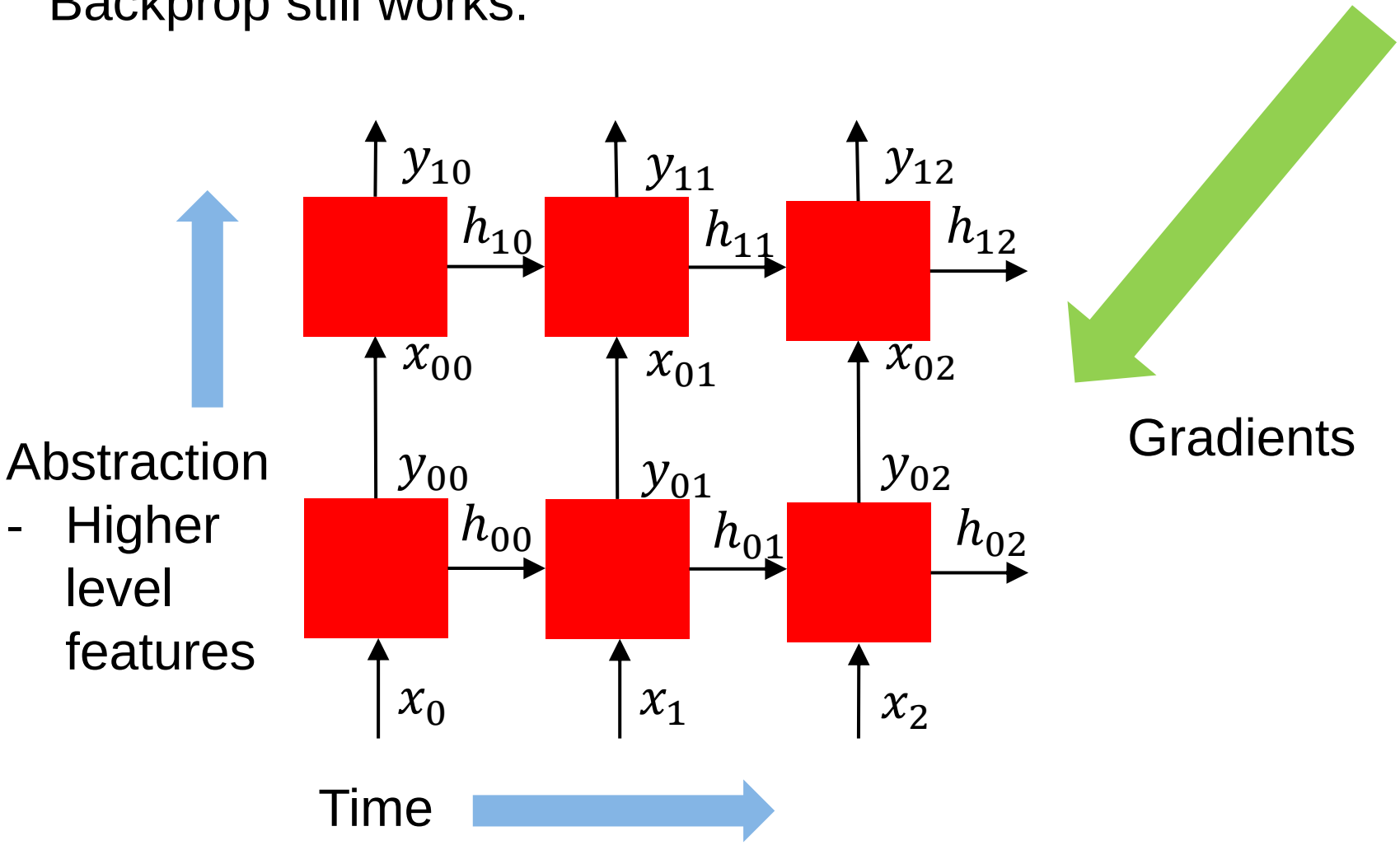
RNN structure

Backprop still works:



RNN structure

Backprop still works:



Recurrent Neural Network

AMRITA
VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY

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

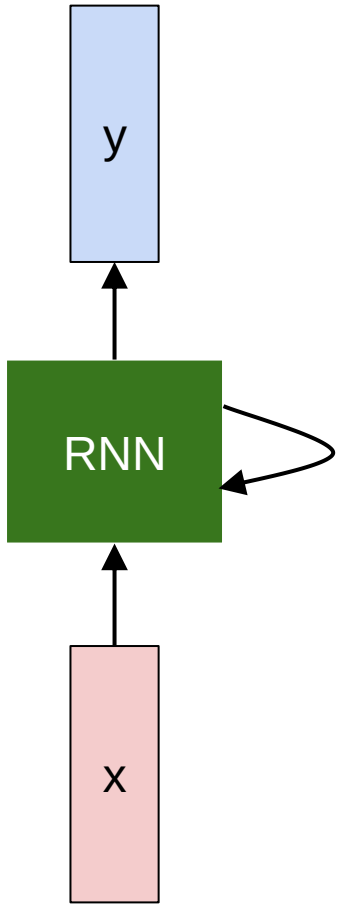
$$h_t = f_W(h_{t-1}, x_t)$$

new state

some function with parameters W

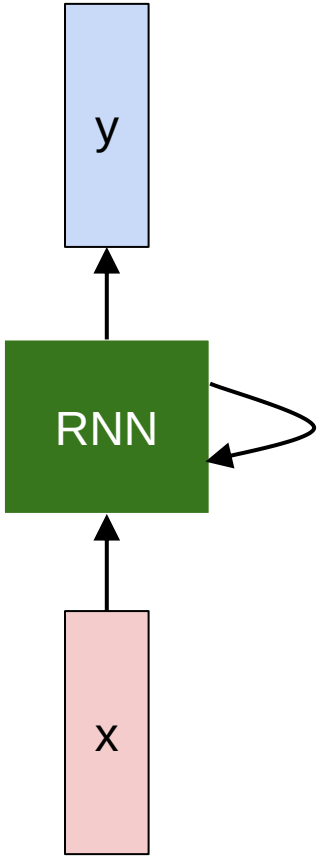
old state

input vector at some time step



Recurrent Neural Network

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

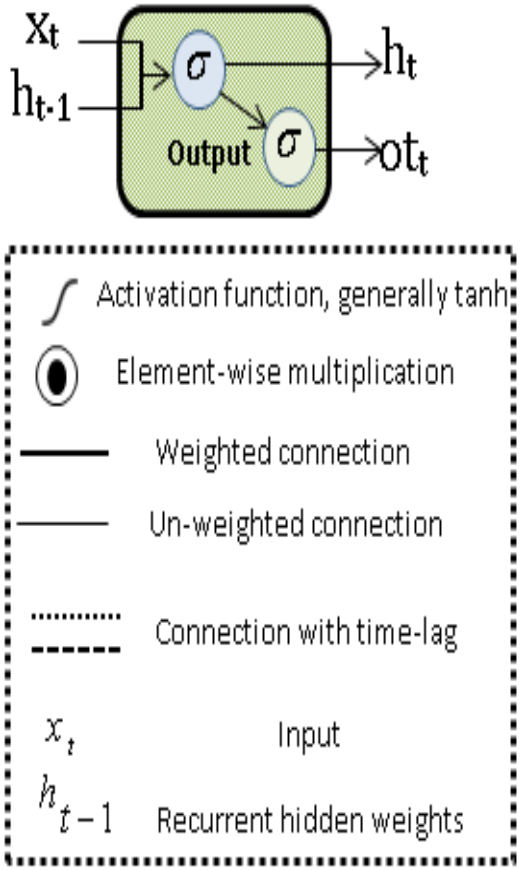


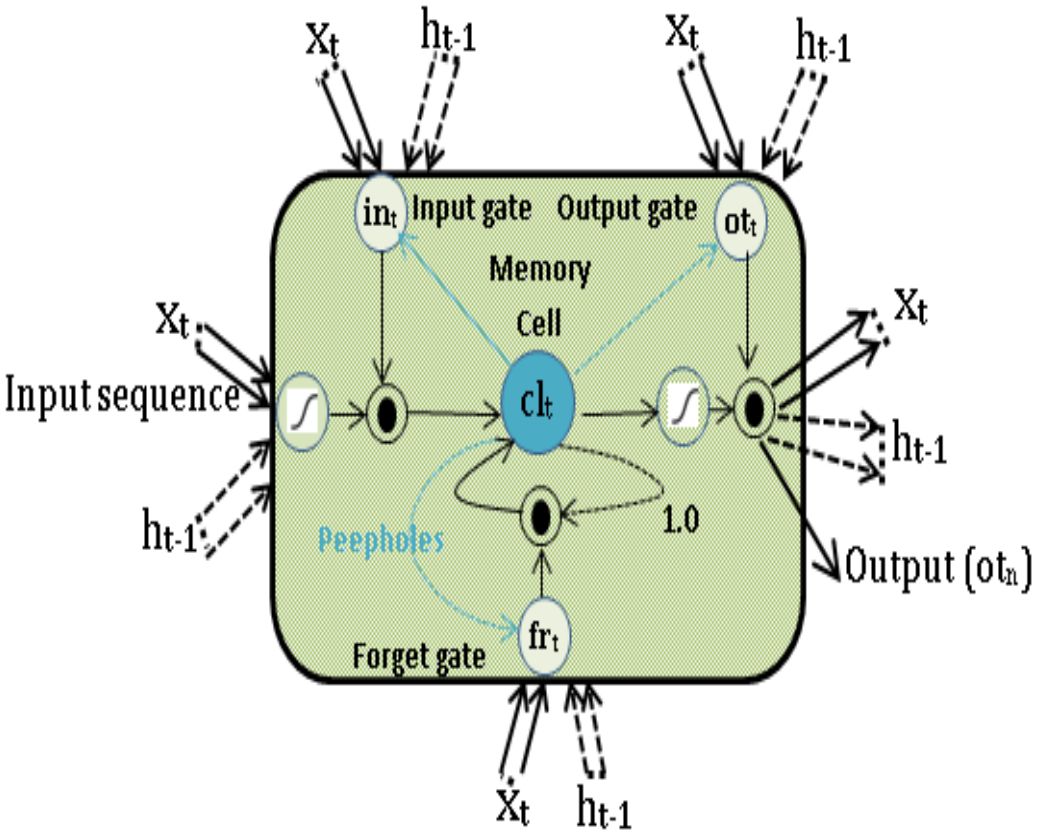
$$h_t = f_W(h_{t-1}, x_t)$$

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

$$y_t = W_{hy}h_t$$

Long short-term memory





Long short-term memory

AMRITA
VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY

$$x_t, h_{t-1}, cl_{t-1} \rightarrow h_t, cl_t$$

$$in_t = \sigma(w_{xin} x_t + w_{hin} h_{t-1} + w_{clin} cl_{t-1} + b_{in})$$

$$fr_t = \sigma(w_{xfr} x_t + w_{hifr} h_{t-1} + w_{clfr} cl_{t-1} + b_{fr})$$

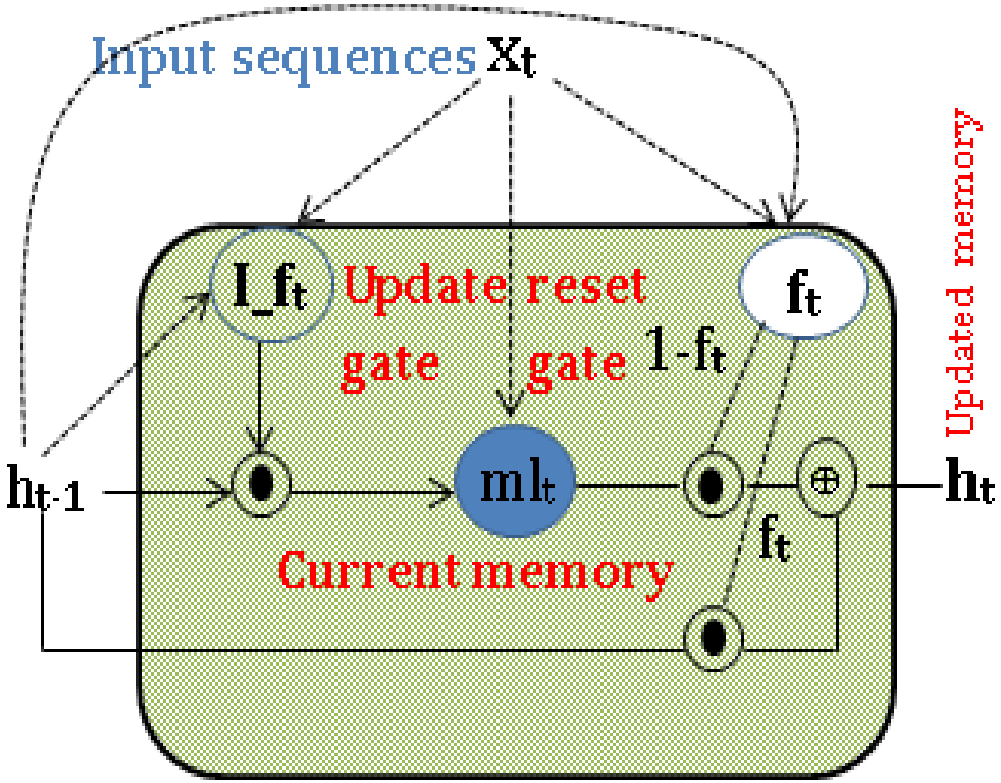
$$cl_t = fr_t \odot cl_{t-1} + in_t \odot \tanh(w_{xcl} x_t + w_{hcl} hi_{t-1} + b_{cl})$$

$$ot_t = \sigma(w_{xot} x_t + w_{hot} hi_{t-1} + w_{clot} cl_t + b_{ot})$$

$$h_t = ot_t \odot \tanh(cl_t)$$

Gated Recurrent Unit

Gated recurrent unit (GRU) is an alternative to LSTM networks.
Formulae shows, unlike LSTM memory cell with a list of gates (input, output and forget), GRU only consist of gates (update and forget) that are collectively involve in balancing the interior flow of information of the unit.



Gated Recurrent Unit

AMRITA
VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY

$$x_t, h_{t-1} \rightarrow h_t$$

$$in\_fr_t = \sigma(w_{xin\_fr} x_t + w_{hiin\_fr} h_{t-1} + b_{in\_fr}) \quad \text{(Update gate)}$$

$$fr_t = \sigma(w_{xfr} x_t + w_{hifr} h_{t-1} + b_{fr}) \quad \text{(Forget or reset gate)}$$

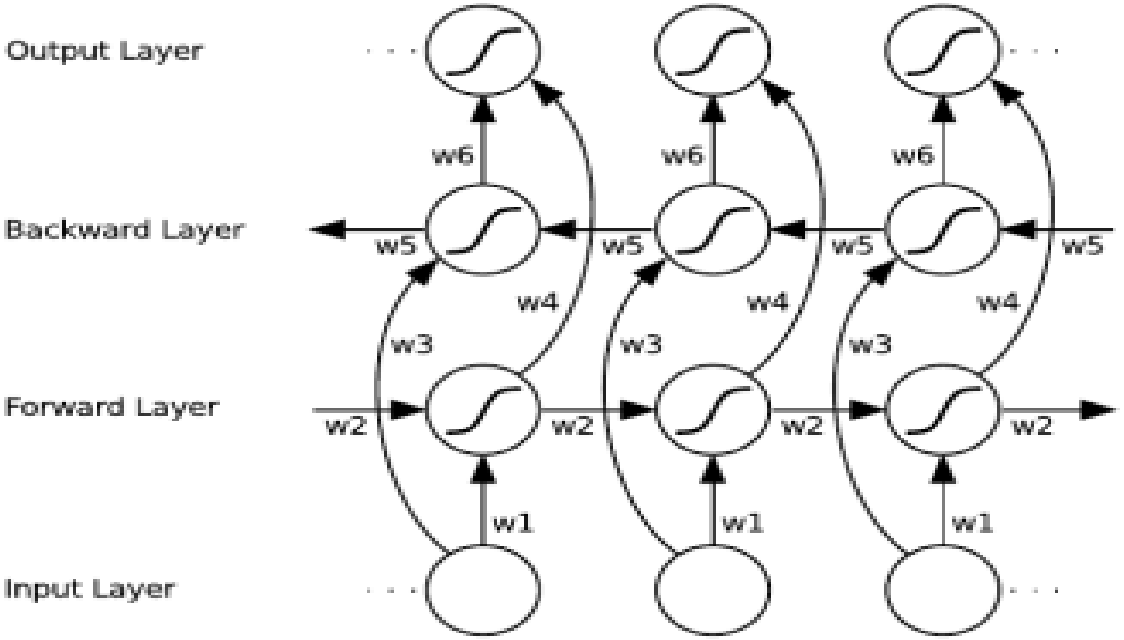
$$cl_t = \tanh(w_{xcl} x_t + w_{hcl}(fr \odot hi_{t-1}) + b_{cl}) \quad \text{(Current memory)}$$

$$h_t = f \odot h_{t-1} + (1-f) \odot cl \quad \text{(Updated memory)}$$

Extensions to LSTM architecture: Bidirectional LSTM

AMRITA VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY

- Only the past information is taken into account in the training of a unidirectional RNN/LSTM
- Bidirectional architecture enables the use of future information
- Implementation with separate Forward-pass and Backward-pass specific layer weights
- Final output computed as the sum of forward and backward layer outputs



Summary

AMRITA
VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY

- RNNs allow a lot of flexibility in architecture design
- RNNs are simple but don’t work very well
- Common to use LSTM or GRU: their additive interactions improve gradient flow
- Backward flow of gradients in RNN can explode or vanish. Exploding is controlled with gradient clipping. Vanishing is controlled with additive interactions (LSTM)
- Better/simpler architectures are a hot topic of current research
- Better understanding (both theoretical and empirical) is needed.

Case studies

AMRITA
VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY

Bio-medical Applications

ECG Analysis

AMRITA
VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY

•An electrocardiogram or ECG signal is a graphical representation which captures the electrical potential changes of the heart

• Electrocardiograph machine is used to obtain ECG signal by capturing the signal through electrodes placed on specific locations on skin of the human body

• For over a period of time, the electrical activity is measured from these leads.

ECG Analysis

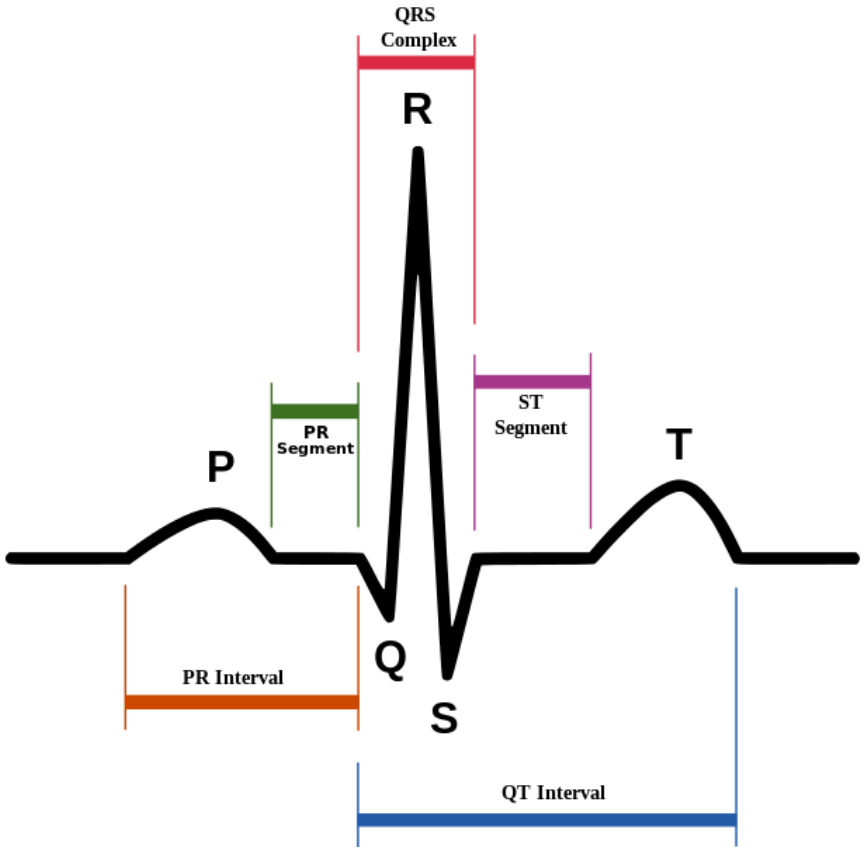
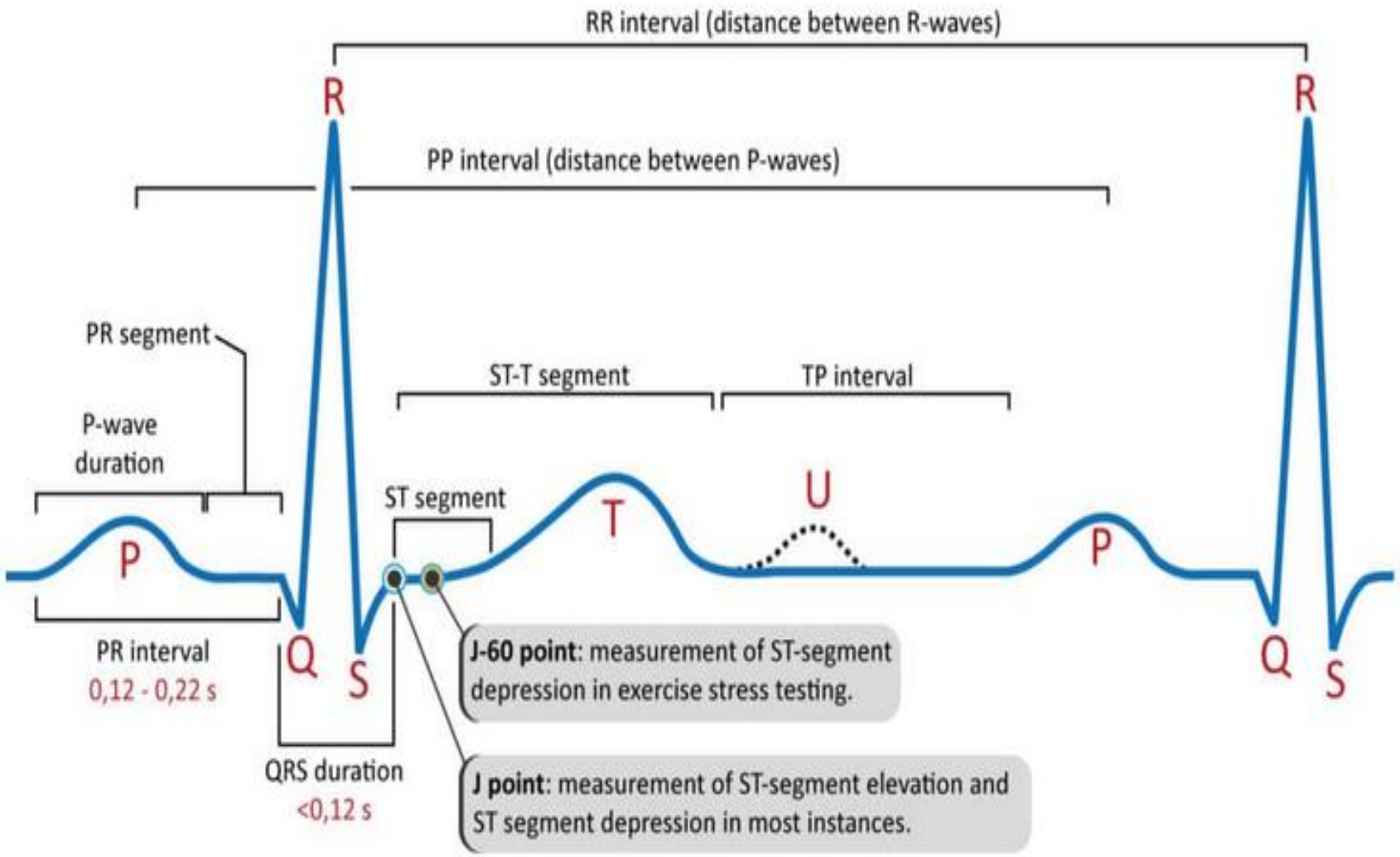


Figure: ECG signal corresponding to normal sinus rhythm

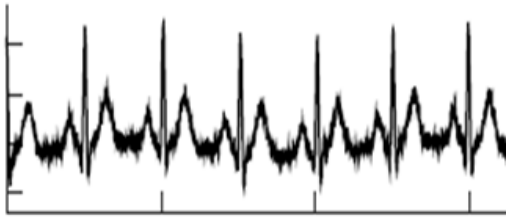
ECG Analysis



ECG Analysis

- P wave occurs due to atrial depolarization, QRS complex due to ventricular depolarization and T wave due to ventricular repolarization.
- QRS complex is the most significant and distinctive feature of ECG used to indicate the presence of cardiac cycle.
- After T, U wave is a small rounded upright wave representing repolarization of Purkinje fibers.
- The intervals that occur in ECG wave are PR Interval and QT Interval.
- PR interval is measured from beginning of P wave to start of QRS complex. It measures travel time of depolarization wave from atria to ventricles.
- QT interval is beginning of QRS complex to end of T wave. It reflects total ventricular activity

Conventional ECG Analysis



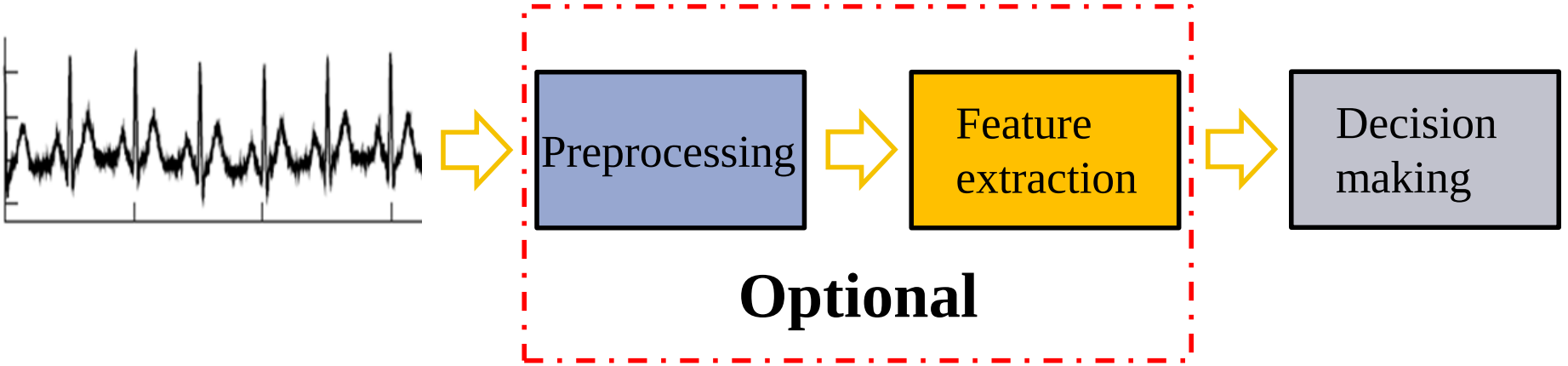
Preprocessing → Feature extraction → Decision making

Noises - PLI, muscle artifacts, motion artifacts, baseline wandering and other external interferences

Features - Slope, width and amplitude of QRS complexes area under the waves etc

AMRITA VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY

Deep Intelligent ECG Analysis



ECG Analysis

AMRITA VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY

Real-time detection of Atrial Fibrillation from Short time single lead ECG traces using Recurrent neural networks

- Atrial fibrillation (AF) is a disorder of the functioning of the heart's electrical system that is characterized by the irregular/rapid beating of the heart [1].
- Atrial fibrillation (AF) is the predominant type of cardiac arrhythmia affecting more than 45 Million individuals globally.
- It is one of the leading contributors of strokes and hence detecting them in real-time is of paramount importance for early intervention.

Proposed Method

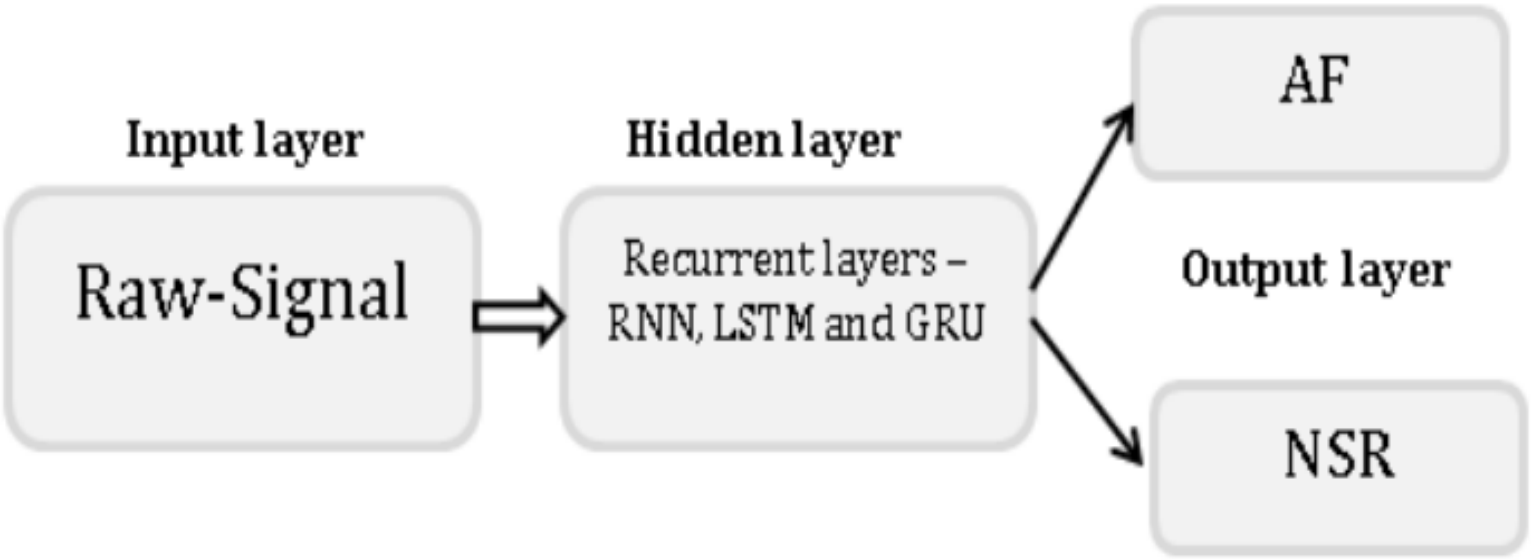
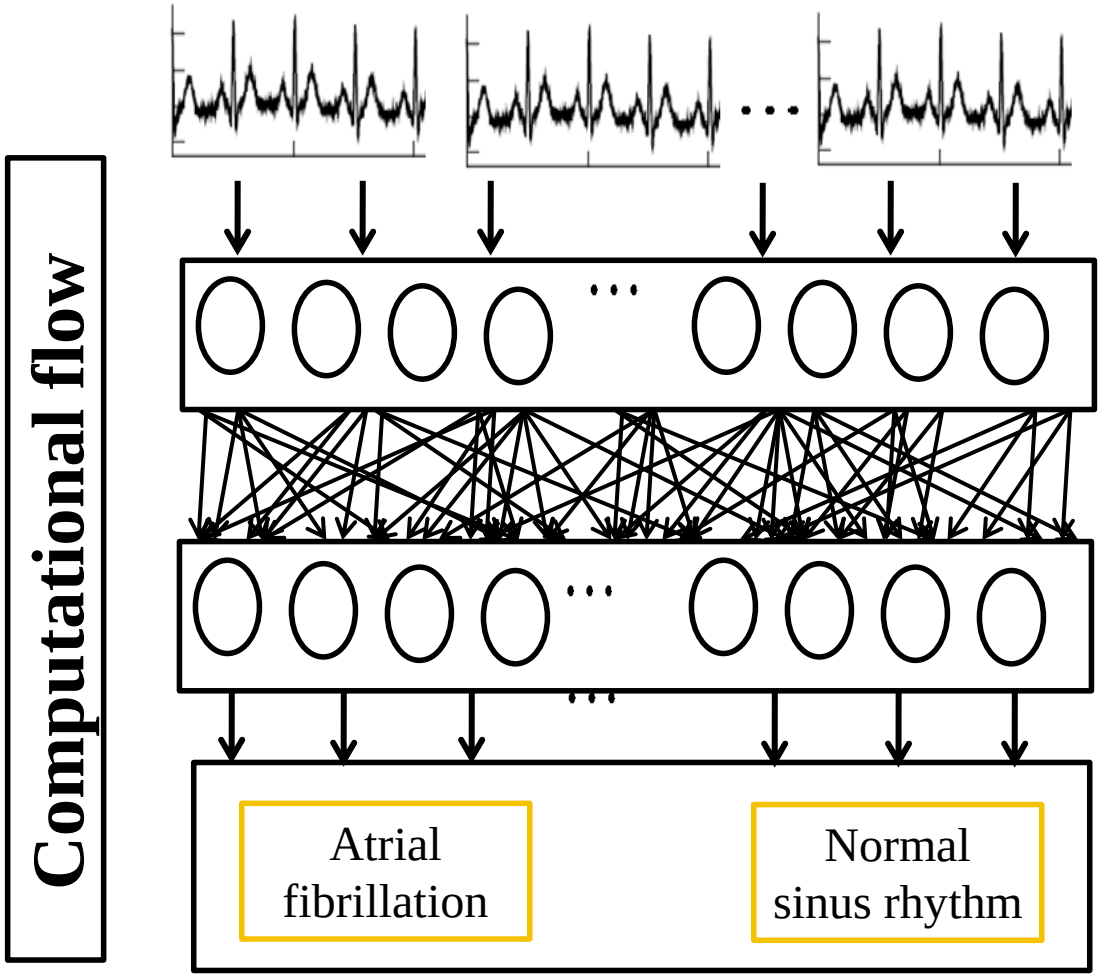


Figure: Architecture of proposed system for normal sinus rhythm and atrial fibrillation.

ECG Analysis



Raw ECG signals

LSTM recurrent layer with 64 memory blocks

Dense layer with sigmoid activation function

Predicted classes

AMRITA VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY

Description of dataset

Name of signal	Number of signals
AF	25
NSR	25

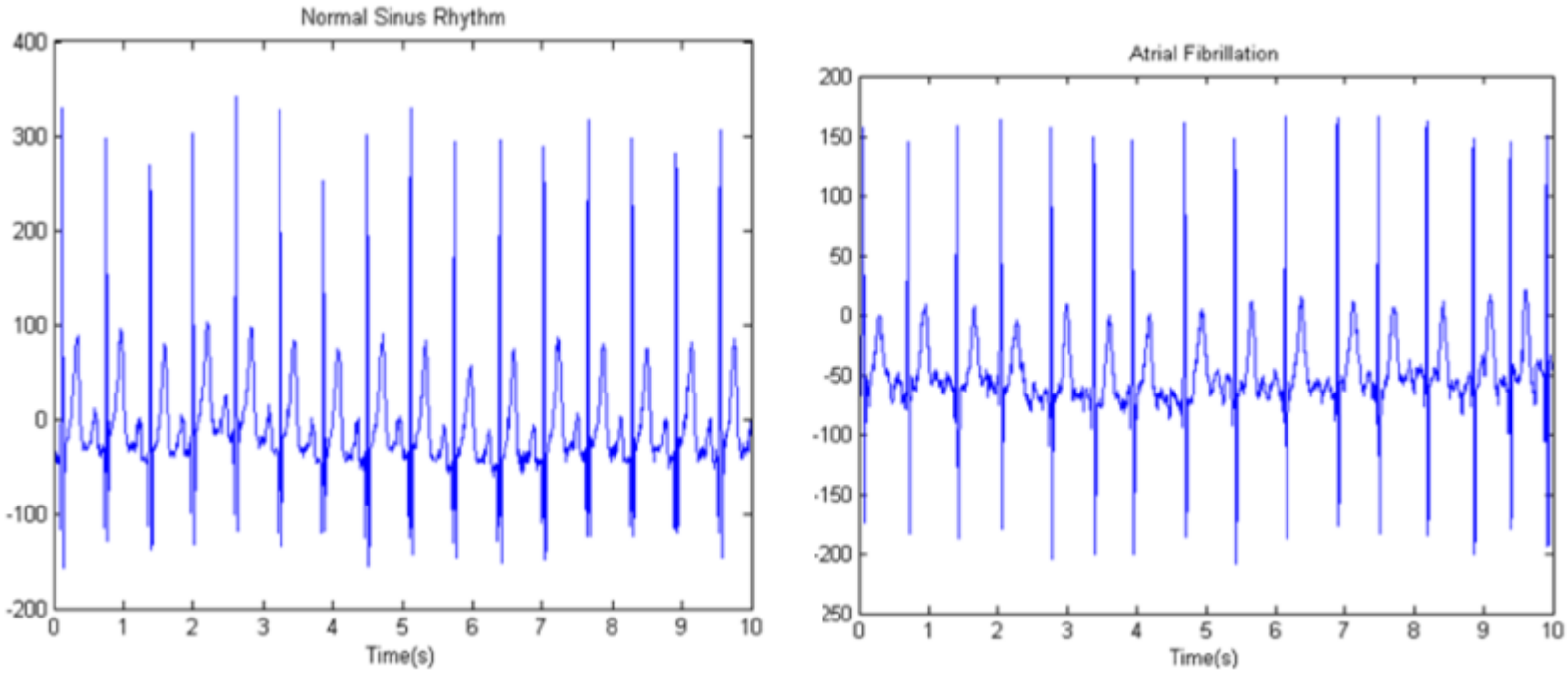


Figure: (a) A single lead ECG wave of normal sinus rhythm,(b) A single lead ECG wave with atrial fibrillation

Results

Algorithm	Accuracy	Precision	Recall	F-score
RNN	0.95	1.00	0.889	0.941
LSTM	1.00	1.00	1.00	1.00
GRU	1.00	1.000	1.00	1.00

The proposed method is considered as more accurate in real-time ECG classification because it doesn't rely on any feature engineering mechanisms.

[1] Sujadevi. V. G, Soman. K. P, Vinaykumar. R, "Real-Time Detection of Atrial Fibrillation from Short Time Single Lead ECG Traces Using Recurrent Neural Networks"- Springer AISC series proceedings of the International Symposium on Intelligent Systems Technologies and Applications 2017.

Sleep Apnea Diagnosis using Deep Learning

- A potentially serious sleep disorder in which breathing repeatedly stops and starts (interruption of breath during sleep).
- Untreated prolonged sleep apnea is directly related to atrial fibrillation, which could later lead to serious conditions such as heart failure and stroke
- Sleep apnea is a medically significant disease condition affecting as much as 24% of men and 9% of women in US population.

Description of dataset

- Physionet Computing in Cardiology (CinC) Sleep Apnea Challenge database.
- 8-hour ECG readings from 35 patients
- It includes three different groups; class A with 20 patient records having very severe sleep apnea incidence, class B with 5 patient records with borderline apnea incidence, and class C with 10 patient records that have less than 5% apnea annotated signals.
- Furthermore, for analyzing the accuracy of sleep apnea classification among arrhythmia patients we obtained 45 ambulatory ECG recordings from MIT BIH Arrhythmia database

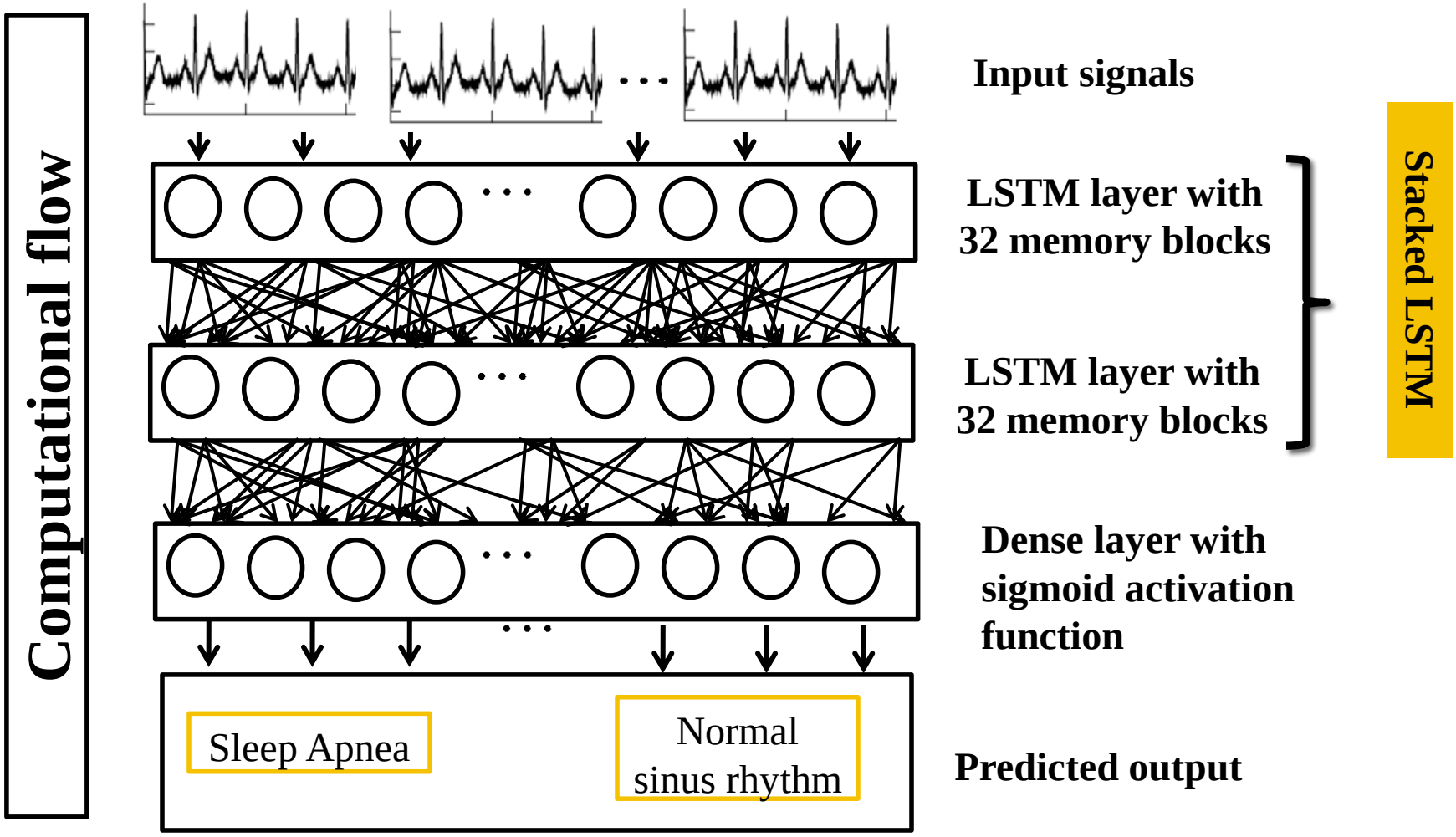
Proposed Method

- The records are taken as length of 60 sec.

- Raw data are processed to extract instantaneous heart rate (IHR), which helps to identify the heart rate variability (HRV) and blood oxygen saturation (SpO2)

- Trained and tested using LSTM/RNN network

ECG Analysis



AMRITA VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY

Results

TABLE I. ACCURACY AND F1 SCORE ON TEST DATASETS.

Dataset	Apnea minutes	Non-Apnea minutes	Accuracy of classification	F1 score
Class A&C	3060	2622	1.0	1.0
Class B	198	1897	1.0	1.0
Arrhythmia	0	1242	0.999	0.999

Pathinarupothi RK, Vinaykumar R, Rangan E, Gopalakrishnan E, Soman KP. Instantaneous heart rate as a robust feature for sleep apnea severity detection using deep learning. In Biomedical & Health Informatics (BHI), 2017 IEEE EMBS International Conference on 2017 Feb 16 (pp. 293-296). IEEE.

Results

TABLE II. ACCURACY, PRECISION AND RECALL OF STACKED LSTM-RNN USING SPO2, IHR AND COMBINATION OF BOTH

Parameter	Accuracy (%)	Precision (%)	Recall (%)
SpO2	95.5	99.2	92.9
IHR	89.0	82.4	99.4
SpO2-IHR	92.1	99.5	84.7

Pathinarupothi RK, Rangan ES, Gopalakrishnan EA, Vinaykumar R, Soman KP. Single Sensor Techniques for Sleep Apnea Diagnosis Using Deep Learning. InHealthcare Informatics (ICHI), 2017 IEEE International Conference on 2017 Aug 23 (pp. 524-529). IEEE.

ECG Analysis

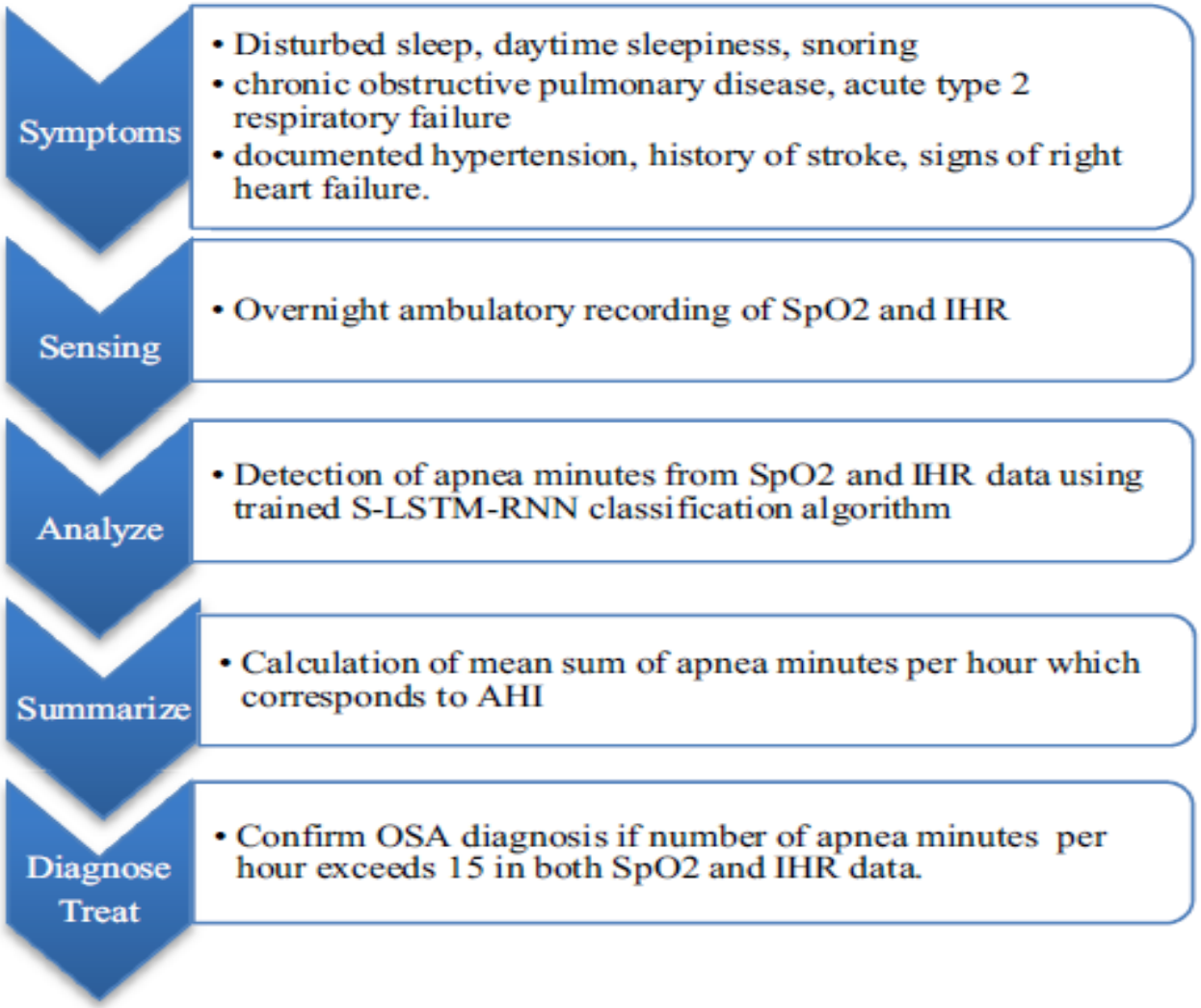
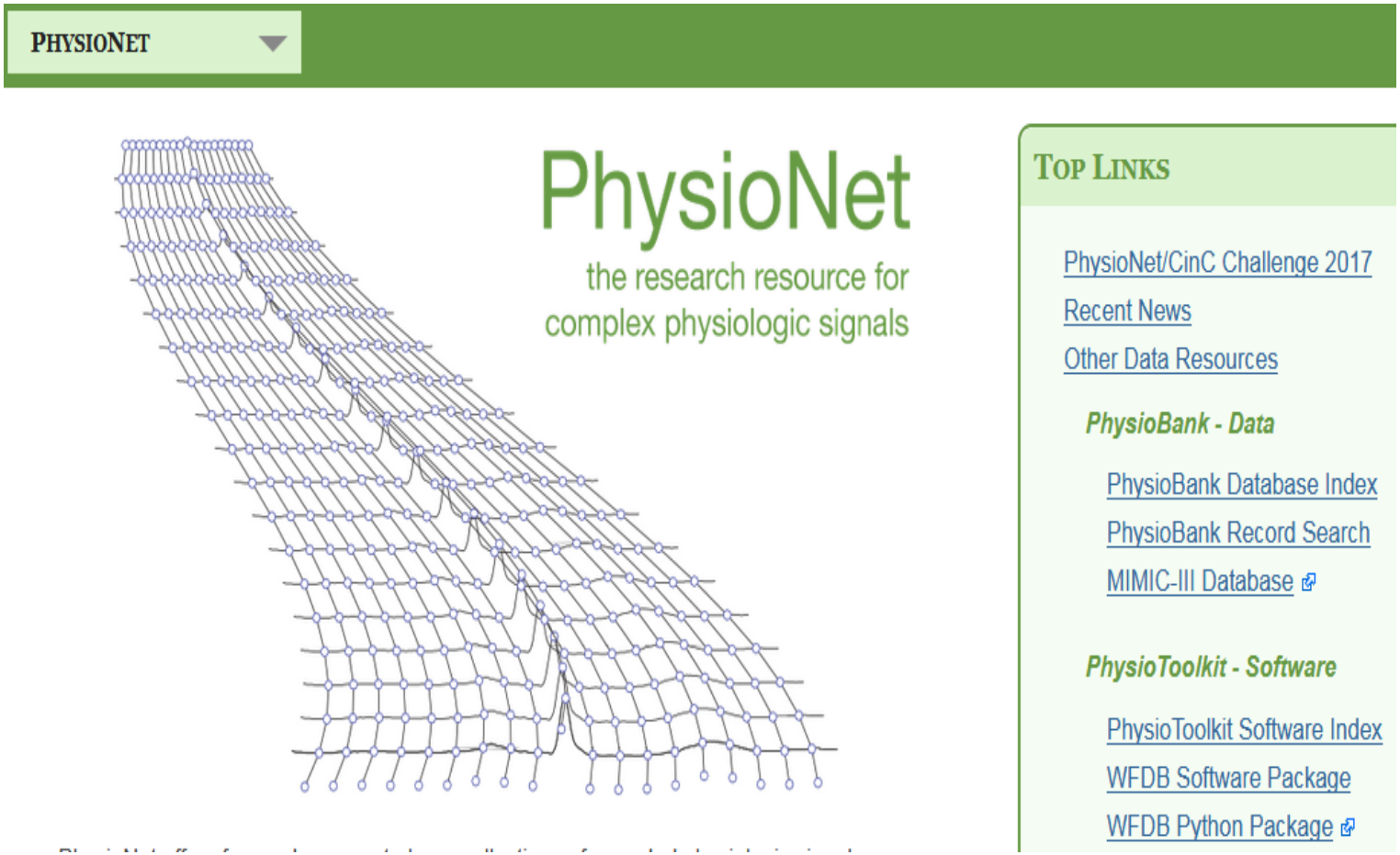


Figure 9: LSTM based new clinical diagnosis protocol for OSA

ECG Analysis

AMRITA VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY

https://physionet.org/



Anomaly detection in Phonocardiogram employing Deep learning

- Phonocardiogram (PCG) is the electronic recording of heart sounds and murmurs
- Detecting abnormal heart sounds by algorithms is important for remote health monitoring and other scenarios where having an experienced physician is not possible
- Detecting anomalies in heart sounds and murmurs using Deep-learning algorithms on well-known Physionet 2016 Dataset
- RNN, LSTM, GRU, B-RNN, B-LSTM and CNN
- We achieved 80% accuracy in CNN 3 layer Deep learning model on the raw signals without performing any preprocessing methods
- To our knowledge this is the highest reported accuracy that employs analyzing the raw PCG data

ECG Analysis

AMRITA VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY

- Sounds of muscles and other artifacts comes as noise to the PCG data generated by hemodynamics
- signal processing that combines wavelet packet transform and singular value decomposition (SVD) for de-noising
- Machine learning methods are being used to identify the anomalies in the signal data
- Machine learning methods relies on the feature engineering and deposing mechanisms
- Deep learning is a new filed of machine learning which can learn the patterns by taking the raw input signals

Proposed Method

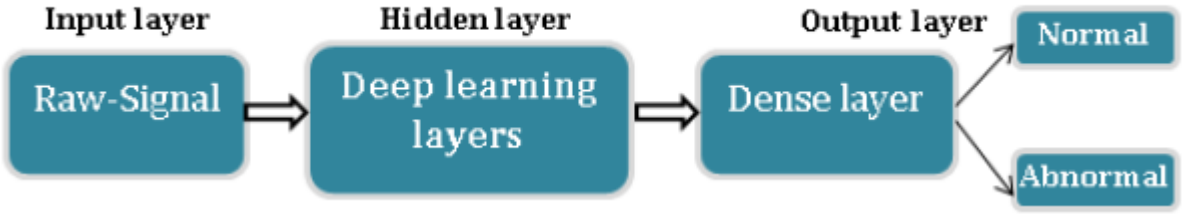


Figure 1. Architecture of proposed system (inner units and their connections are not shown)

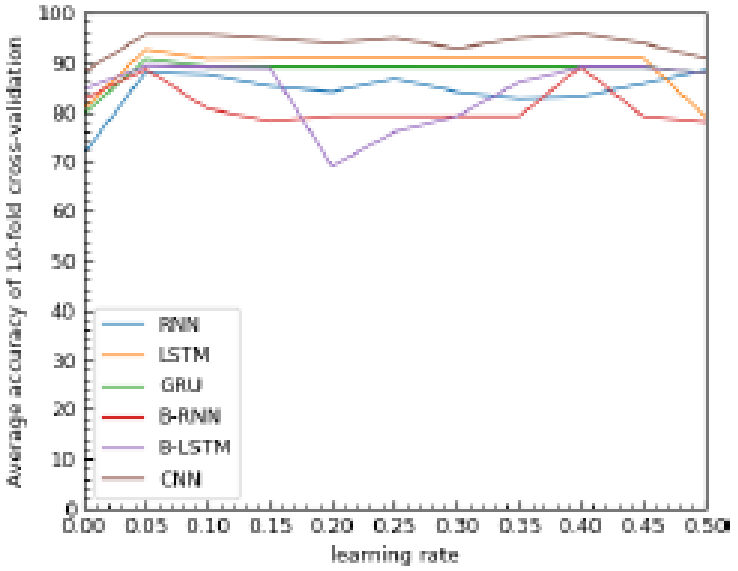
Description of the data set and Results

- As part of 2016 PhysioNet/CinC Challenge, the database of heart recordings has been provided to participants and made available to the researchers for further enhancement in classifying the heart recordings either as normal or abnormal.

Name of Signal	Number of Signals		Name of Signal	Number of Signals	
	Normal	Abnormal		Normal	Abnormal
Training-a	117	292	validation-a	40	40
Training-b	386	104	validation-b	49	49
Training-c	7	24	validation-c	4	3
Training-d	27	28	validation-d	5	5
Training-e	1958	183	validation-e	53	53
Training-f	80	34	validation-f	-	-
Total	2575	665	Total	151	150

Hyper-parameter tuning

- To find suitable parameter for RNN, LSTM, GRU, B-RNN, B-LSTM, we used with one hidden layer, 32, 64 and 128 units in RNN and 32, 64 and 128 memory blocks in LSTM.
- 3 trails of experiments, each 500 epochs
- 64 units/memory blocks has shown highest accuracy in 10-fold cross-validation for most of the Deep learning architectures
- CNN with number of Filters 64 and filter length 3 has attained highest accuracy in 10-fold cross-validatio



CNN Tuning

- To alleviate the cost of training with CNN network, we apply fast fourier transform (FFT) on the raw signals to convert the signal into a frequency domain.
- High pass Butterworth Filter is used to remove the noise above 240 beats per minute (4Hz).
- Again, we apply FFT on the filtered signal to transform into to it's approximate frequency domain.
- Finally, we pass the Fourier coefficients into CNN network.

ECG Analysis

AMRITA VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY

Test Results

Algorithm	Accuracy	Precision	Recal	F-measure
RNN 1 layer	0.638	0.626	0.800	0.702
RNN 2 layer	0.653	0.672	0.683	0.678
RNN 3 layer	0.680	0.737	0.622	0.675
RNN 4 layer	0.688	0.674	0.806	0.734
LSTM 1 layer	0.688	0.736	0.606	0.665
LSTM 2 layer	0.685	0.710	0.694	0.702
LSTM 3 layer	0.700	0.734	0.689	0.711
LSTM 4 layer	0.715	0.747	0.706	0.726
GRU 1 layer	0.694	0.688	0.783	0.732
GRU 2 layer	0.691	0.713	0.706	0.709
GRU 3 layer	0.703	0.741	0.683	0.711
GRU 4 layer	0.712	0.743	0.706	0.724
B-RNN 1 layer	0.685	0.720	0.672	0.695
B-RNN 2 layer	0.680	0.728	0.639	0.680
B-RNN 3 layer	0.691	0.724	0.683	0.703
B-RNN 4 layer	0.715	0.744	0.711	0.727
B-LSTM 1 layer	0.724	0.740	0.744	0.742
B-LSTM 2 layer	0.709	0.716	0.756	0.735
B-LSTM 3 layer	0.700	0.698	0.772	0.734
B-LSTM 4 layer	0.736	0.741	0.778	0.759
CNN 1 layer	0.703	0.725	0.717	0.721
CNN 2 layer	0.773	0.771	0.773	0.770
CNN3layer	**0.798**	**0.815**	**0.806**	**0.810**

Summary and Future work

- Deep learning based mechanism such as RNN, LSTM and GRU architecture is proposed to detect the Anomaly in PCG data.
- All the deep learning methods have performed well with the raw data, without doing any feature engineering. Noise filtered data with CNN gave the highest accuracy.
- In our experiment CNN has given the best results when compared to other networks. The primary reason is that, it used FFT with high pass Butterworth Filter which removed the noise and obtain information related to frequency domain.
- The reported results can be further enhanced by following hyper parameter tuning mechanism for each deep network architecture.
- And to our knowledge this is the highest accuracy reported so far on raw PCG data anomaly detection

AMRITA
VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY

Thank you

Questions ?

vinayakumarr77@gmail.com
https://vinayakumarr.github.io/
https://sites.google.com/site/vinayakumarr77/

AMRITA
VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY

Hands on session on “Machine learning and Deep learning using Scikit-learn, Tensorflow and Keras”

Software Installation

AMRITA
VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY

- sudo apt-get install libatlas-base-dev gfortran python-dev
- sudo apt-get install python-pip
- sudo pip install --upgrade pip
- sudo pip install numpy
- sudo pip install scipy
- sudo pip install matplotlib
- Sudo pip install seaborn
- sudo pip install scikit-learn
- sudo pip install tensorflow
- sudo pip install theano
- sudo pip install keras
- sudo pip install pandas
- sudo pip install h5py
- sudo pip install jupyter
- sudo pip install ipython

Artificial Intelligence (AI) toolkits

AMRITA
VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY

Scikit-learn - Python library that implements a comprehensive range of machine learning algorithms.

- easy-to-use, general-purpose toolbox for machine learning in Python.
- supervised and unsupervised machine learning techniques.
- Utilities for common tasks such as model selection, feature extraction, and feature selection.
- Built on NumPy, SciPy, and matplotlib.
- Open source, commercially usable - BSD license.

Artificial Intelligence (AI) toolkits

AMRITA
VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY

TensorFlow - library for numerical computation using data flow graphs / deep learning.

- Open source
- By Google
- used for both research and production
- Used widely for deep learning/neural nets
- But not restricted to just deep models
- Multiple GPU Support

Artificial Intelligence (AI) toolkits

AMRITA
VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY

Keras – It is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It was developed with a focus on enabling fast experimentation.

- Allows for easy and fast prototyping (through user friendliness, modularity, and extensibility).
- Runs seamlessly on CPU and GPU.

Supporting Libraries

NumPy

Base N-dimensional array package

SciPy library

Fundamental library for scientific computing

Matplotlib

Comprehensive 2D Plotting

IP[y]: IPython

IPython

Enhanced Interactive Console

Sympy

Symbolic mathematics

pandas

Data structures & analysis

AMRITA
VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY

Hands – on tutorial on supporting libraries

AMRITA
VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY

Hands – on tutorial on classical machine learning algorithms using scikit-learn

AMRITA
VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY

Hands – on tutorial on Tensorflow with Keras