Scope and opportunities in the areas of

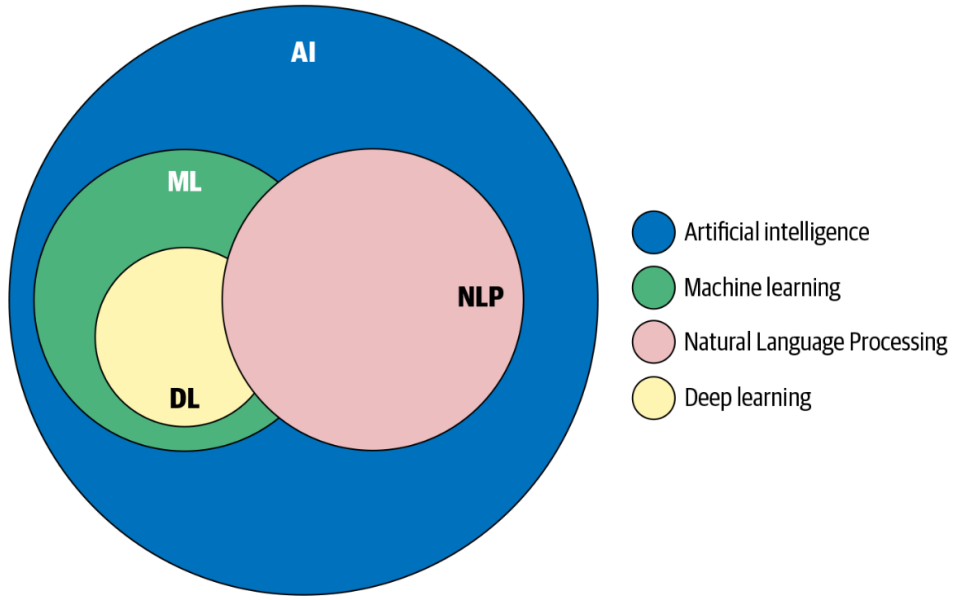
AI-NLP : RNN

Dr. Sarwan Singh

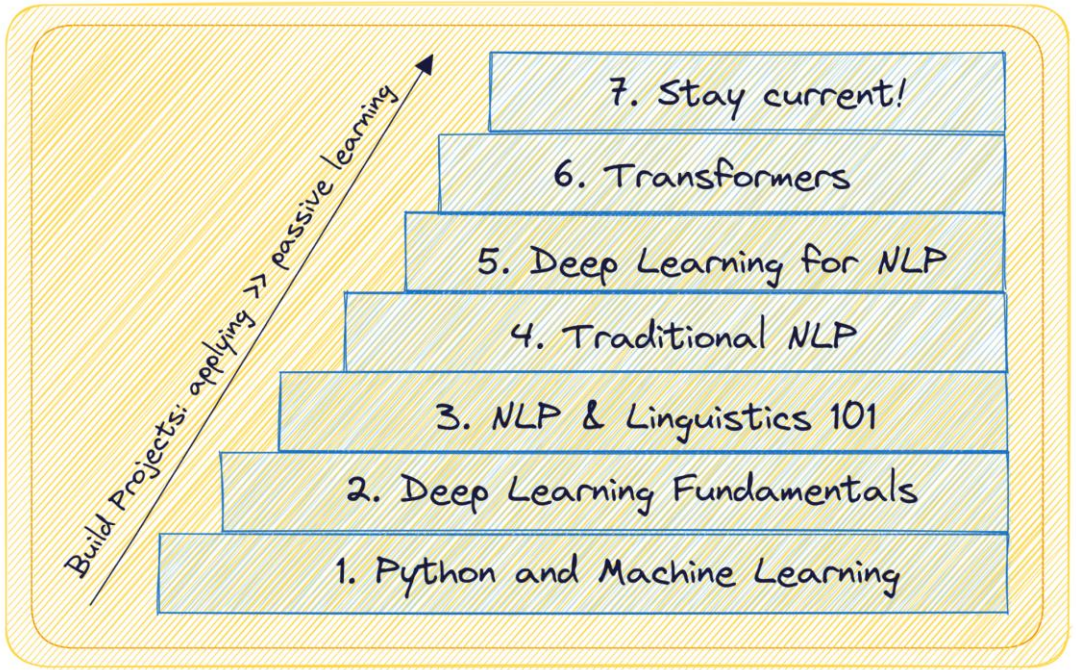
Scientist – D, NIELIT Chandigarh

Special thanks/ references

- Pankaj Gupta PhD Student@university of Munich
- https://aws.amazon.com/



- Step 1: Python and ML fundamentals
- Step 2: Deep learning fundamentals
- Step 3: NLP essential linguistics concepts
- Step 4: Traditional NLP techniques
- Step 5: Deep learning for NLP
- Step 6: NLP with transformers
- Step 7: Build projects, keep learning, and stay current!



Source : https://www.kdnuggets.com/

Need for Sequential Modeling

Examples of Sequence data

- Speech Recognition
- Machine Translation
- Language Modeling
- Named Entity Recognition
- Sentiment Classification
- Video Activity Analysis

Input Data

Hello, I am Sarwan.

Recurrent neural ? based ? model

Sarwan lives in Chandigarh

There is nothing to like in this movie.

Output

This is RNN

Hallo, ich bin Sarwan

हैलो, म

network

language

Sarwan lives in **Chandigarh**

person ***location***

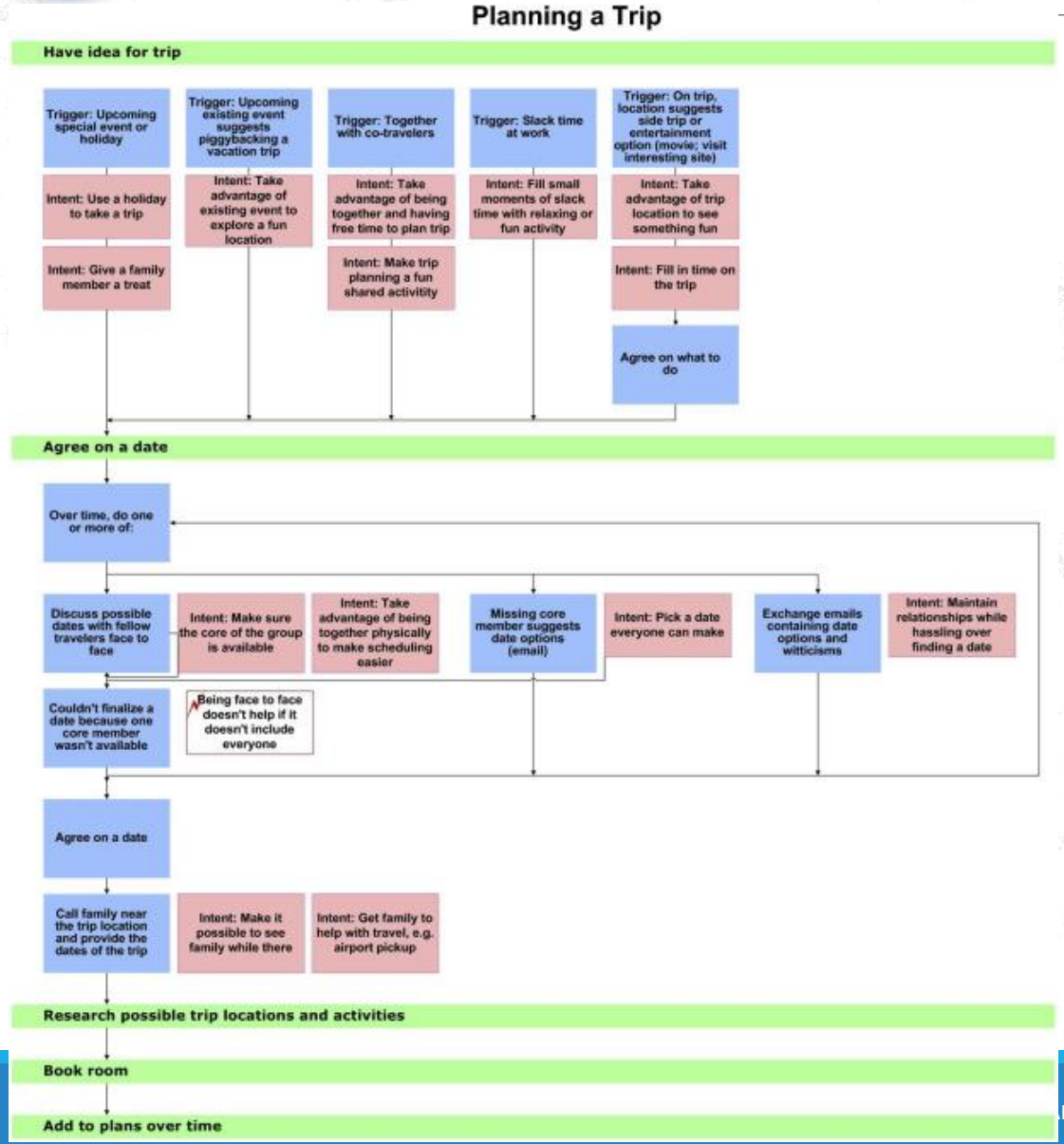
Punching

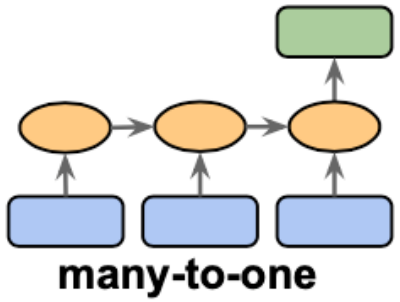
Need for Sequential Modeling

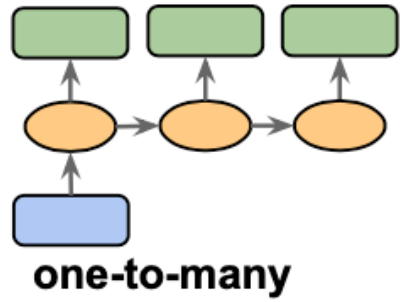
- A Sequence Model is defined as a structured representation of a task, detailing the activities, steps, intents, and breakdowns involved in completing the task. It helps in designing a more efficient way to fulfill the primary intent of the task and improve the flow between different activities and steps.
- A sequencing model is a machine learning model that processes inputs as a sequence, with each data point conditioned on the ones that come before it. This allows the model to learn complex time-based patterns, such as trends, seasonality, and long-range dependencies

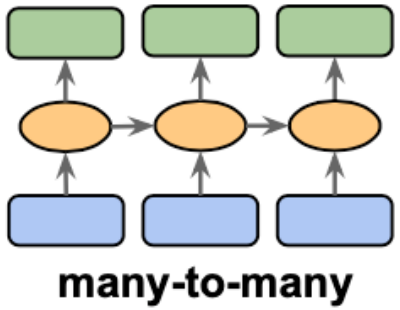
Sequence models are used in a variety of applications, including:

- **Natural Language Processing (NLP)**: Sequence models are used for language translation, text generation, and sentiment classification.
- **Speech recognition**: Sequence models are used to convert spoken language into textual form.
- **Music generation**: Sequence models are used to generate music.
- **Forecasting stocks**: Sequence models are used to forecast stocks.
- Sequence models are different from traditional models because they are designed to manage variable-length sequences and capture intricate dependencies between elements. They do this by maintaining a "state" or "memory" across inputs, allowing the model to remember previous inputs and use this information to influence future predictions.









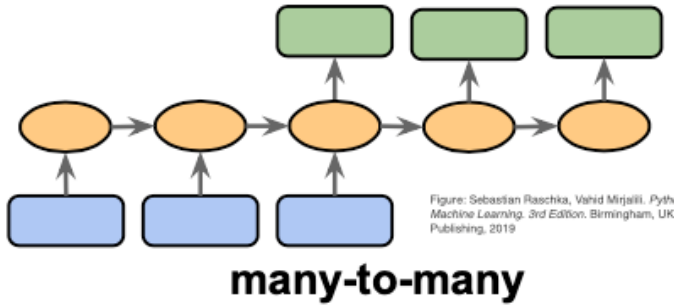
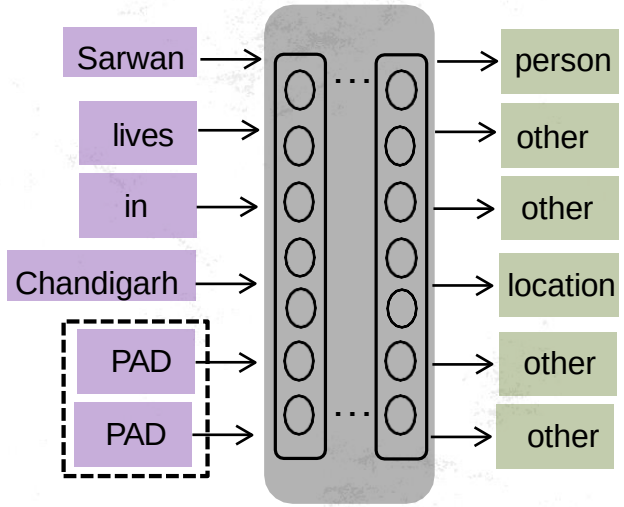


Figure: Sebastian Raschka, Vahid Mirjalili. *Python Machine Learning. 3rd Edition*. Birmingham, UK: Packt Publishing, 2019

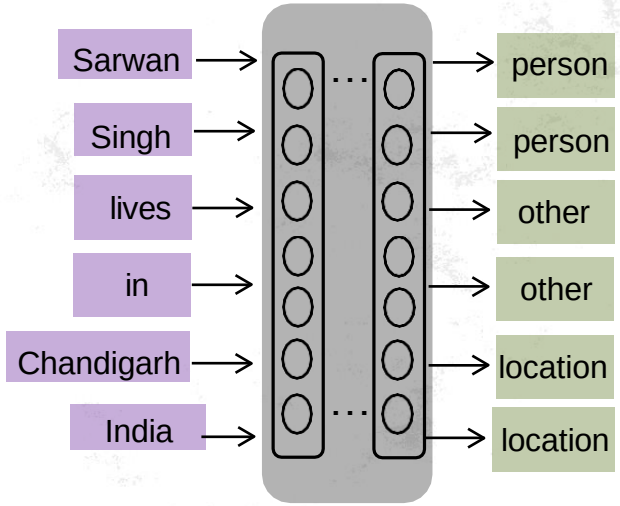
Inputs, Outputs can be different lengths in different examples

Example:

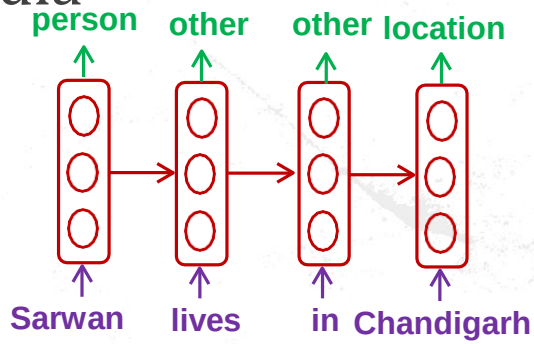
- Sentence1: Sarwan lives in Chandigarh
- Sentence2: Sarwan Singh lives in Chandigarh India
- FF – Feed Forward network



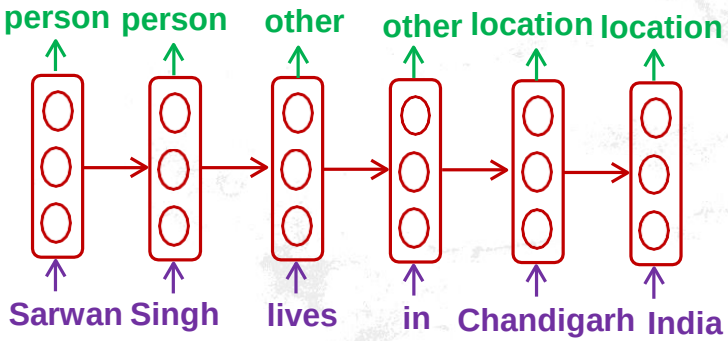
FF-net / CNN



FF-net / CNN



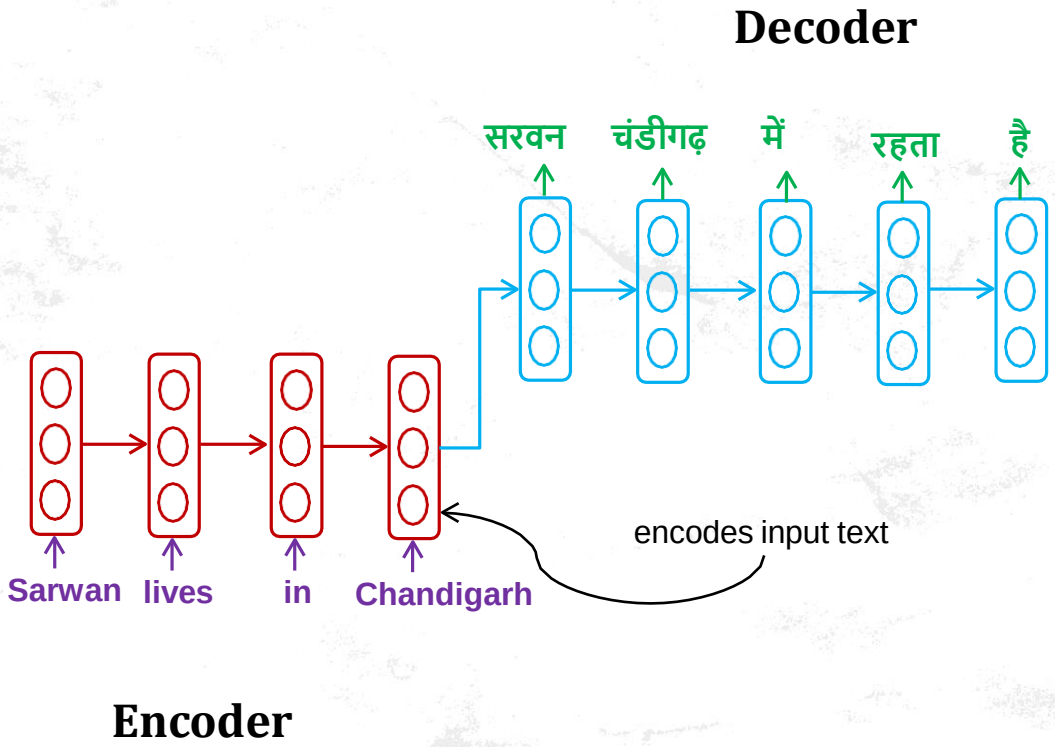
Models variable length sequences



Sequential model: RNN

Encoder – Decoder

- Machine Translation: Different Input and Output sizes, incurring sequential patterns



CNN vs RNN

- Convolutional vs Recurrent Neural Networks

RNN

- perform well when the input data is interdependent in a sequential pattern
- correlation between previous input to the next input
- introduce bias based on your previous output

CNN/FF-Nets

- all the outputs are self dependent
- Feed-forward nets don't remember historic input data at test time unlike recurrent networks.

Memory less vs Memory Networks

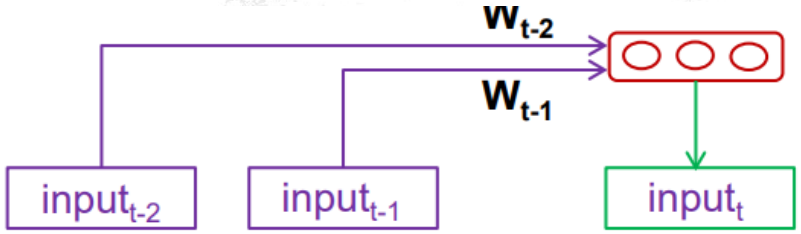
Memory-less Models

Autoregressive models:

- Predict the next input in a sequence from a fixed number of previous inputs using "delay taps".

Feed-forward neural networks:

- Generalize autoregressive models by using non-linear hidden layers.



Memory Networks

- possess a dynamic hidden state that can store long term information, e.g., RNNs.

Recurrent Neural Networks:

- RNNs are very powerful, because they combine the following properties-
- Distributed hidden state: can efficiently store a lot of information about the past.
- Non-linear dynamics: can update their hidden state in complicated ways
- Temporal and accumulative: can build semantics, e.g., word-by-word in sequence over time

Long Term and Short Dependencies

Short Term Dependencies

- need recent information to perform the present task.
- For example in a language model, predict the next word based on the previous ones
- “the clouds are in the ?” ‘sky’ “the clouds are in the **sky**”
- Easier to predict ‘sky’ given the context, i.e., short term dependency

Long Term Dependencies

- Consider longer word sequence “I grew up in Punjab…….. I speak fluent Punjabi.”
- Recent information suggests that the next word is probably the name of a language, but if we want to narrow down which language, we need the context of Punjab, from further back.

Foundation of Recurrent Neural Networks

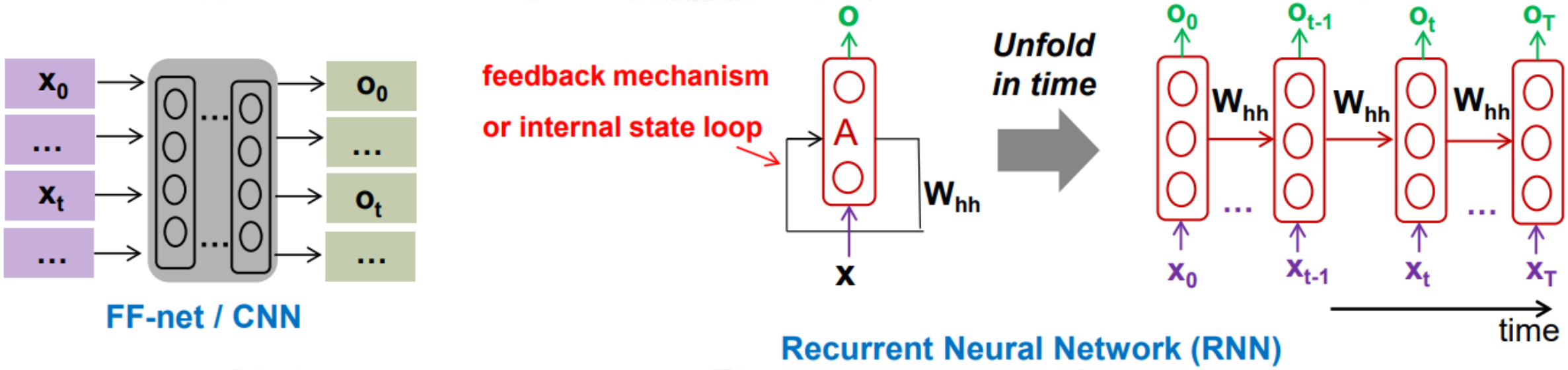
Goal

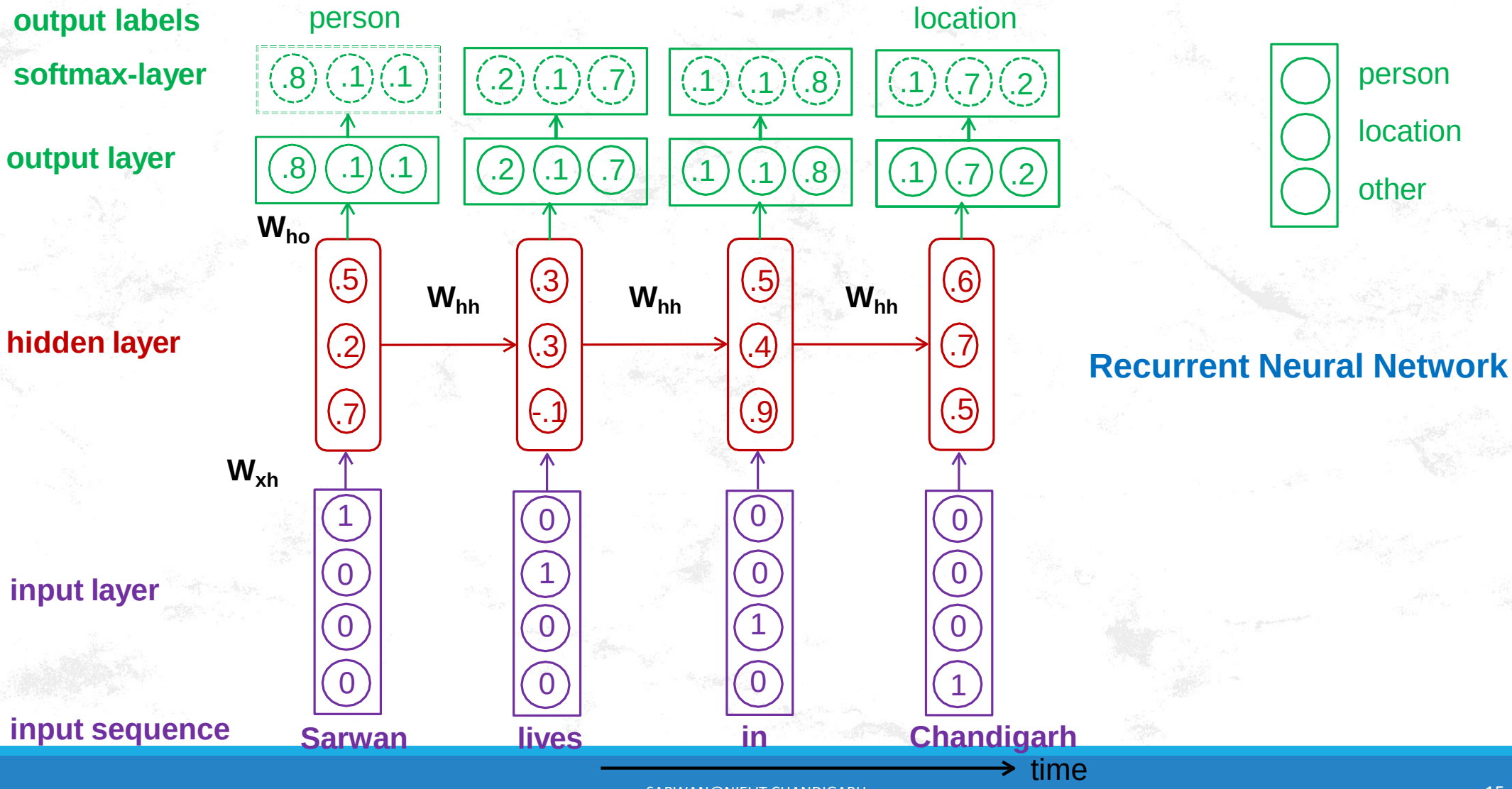
- model long term dependencies
- connect previous information to the present task
- model sequence of events with loops, allowing information to persist

Punching

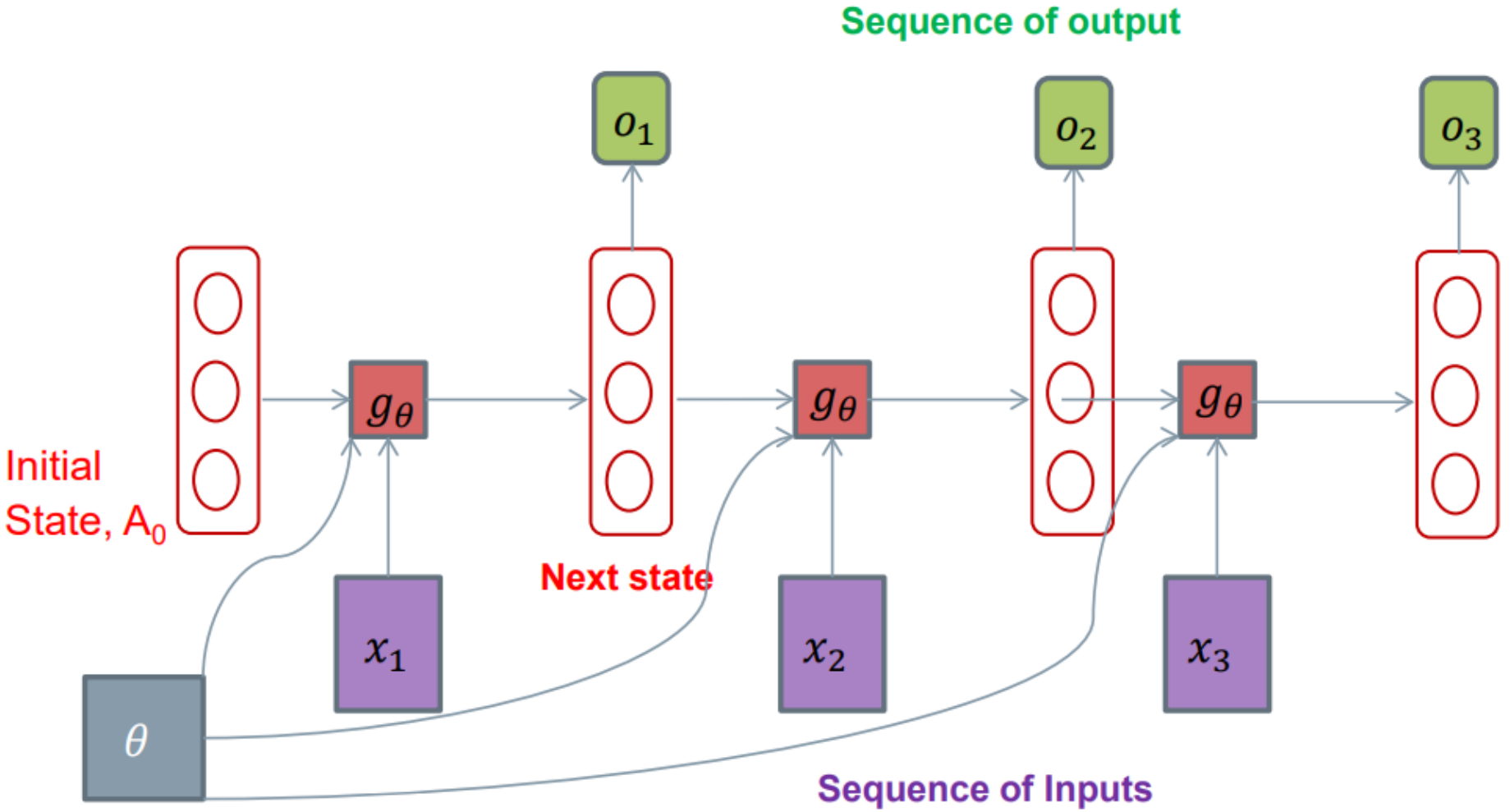
Foundation of Recurrent Neural Networks

- Feed Forward NNets can not take time dependencies into account.
- Sequential data needs a Feedback Mechanism.

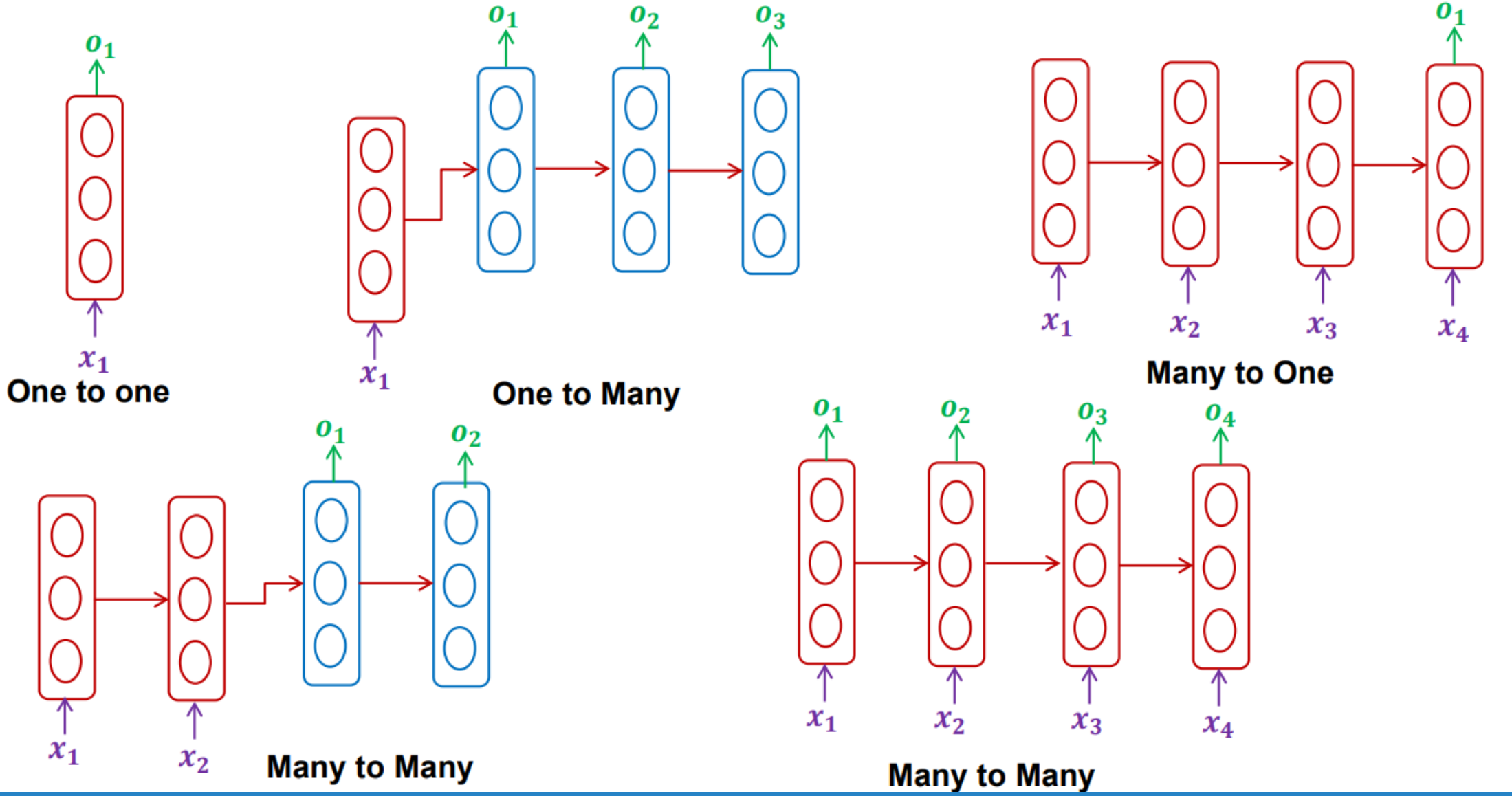



output labels
softmax-layer
output layer
hidden layer
input layer
input sequence
person
location
.8 .1 .1
.2 .1 .7
.1 .1 .8
.1 .7 .2
W_{ho}
W_{hh}
W_{hh}
W_{hh}
W_{xh}
.5 .2 .7
.3 .3 -.1
.5 .4 .9
.6 .7 .5
1 0 0 0
0 1 0 0
0 0 1 0
0 0 0 1
Sarwan
lives
in
Chandigarh
time
person
location
other
Recurrent Neural Network

RNN: Computational Graphs



RNN: Different Computational Graphs

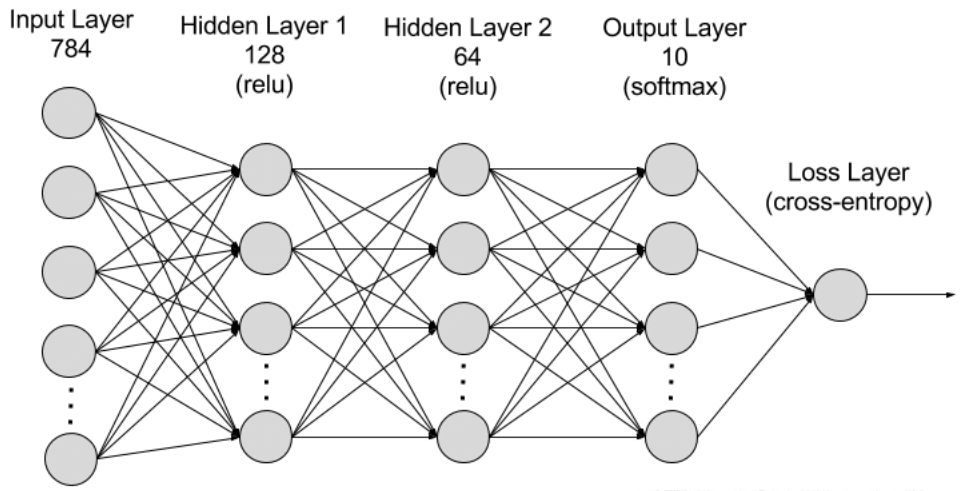


RNN – in simple words

- A recurrent neural network (RNN) is a deep learning model that is trained to process and convert a sequential data input into a specific sequential data output.
- Sequential data is data—such as words, sentences, or time-series data—where sequential components interrelate based on complex semantics and syntax rules.
- An RNN is a software system that consists of many interconnected components mimicking how humans perform sequential data conversions, such as translating text from one language to another.
- RNNs are largely being replaced by transformer-based artificial intelligence (AI) and large language models (LLM), which are much more efficient in sequential data processing.

How does a recurrent neural network work

- RNNs are made of neurons: data-processing nodes that work together to perform complex tasks.
- The neurons are organized as input, output, and hidden layers.
- The input layer receives the information to process, and the output layer provides the result.
- Data processing, analysis, and prediction take place in the hidden layer.



Hidden layer

- RNNs work by passing the sequential data that they receive to the hidden layers one step at a time. However, they also have a self-looping or *recurrent* workflow: the hidden layer can remember and use previous inputs for future predictions in a short-term memory component. It uses the current input and the stored memory to predict the next sequence.
- For example, consider the sequence: *Apple is red.* You want the RNN to predict *red* when it receives the input sequence *Apple is.* When the hidden layer processes the word *Apple*, it stores a copy in its memory. Next, when it sees the word *is*, it recalls *Apple* from its memory and understands the full sequence: *Apple is* for context. It can then predict *red* for improved accuracy. This makes RNNs useful in speech recognition, machine translation, and other language modeling tasks.

Training

- Machine learning (ML) engineers train deep neural networks like RNNs by feeding the model with training data and refining its performance.
- In ML, the neuron's weights are signals to determine how influential the information learned during training is when predicting the output. Each layer in an RNN shares the same weight.
- ML engineers adjust weights to improve prediction accuracy. They use a technique called backpropagation through time (BPTT) to calculate model error and adjust its weight accordingly.
- BPTT rolls back the output to the previous time step and recalculates the error rate. This way, it can identify which hidden state in the sequence is causing a significant error and readjust the weight to reduce the error margin.

Types of Recurrent Neural Networks

RNNs are often characterized by one-to-one architecture: one input sequence is associated with one output. However, you can flexibly adjust them into various configurations for specific purposes. The following are several common RNN types.

One-to-many

- This RNN type channels one input to several outputs. It enables linguistic applications like image captioning by generating a sentence from a single keyword.

Many-to-many

- The model uses multiple inputs to predict multiple outputs. For example, you can create a language translator with an RNN, which analyzes a sentence and correctly structures the words in a different language.

Many-to-one

- Several inputs are mapped to an output. This is helpful in applications like sentiment analysis, where the model predicts customers' sentiments like *positive, negative*, and *neutral* from input testimonials.

Recurrent neural network vs. feed-forward neural network

- Like RNNs, feed-forward neural networks are artificial neural networks that pass information from one end to the other end of the architecture.
- A feed-forward neural network can perform simple classification, regression, or recognition tasks, but it can't remember the previous input that it has processed.
- For example, it forgets *Apple* by the time its neuron processes the word *is*. The RNN overcomes this memory limitation by including a hidden memory state in the neuron.

Recurrent neural network vs. convolutional neural networks

- Convolutional neural networks are artificial neural networks that are designed to process spatial data.
- You can use convolutional neural networks to extract spatial information from videos and images by passing them through a series of convolutional and pooling layers in the neural network.
- RNNs are designed to capture long-term dependencies in sequential data

some variants of RNN

- The RNN architecture laid the foundation for ML models to have language processing capabilities. Several variants have emerged that share its memory retention principle and improve on its original functionality. The following are some examples.
- Bidirectional recurrent neural networks
- Long short-term memory
- Gated recurrent units