Recurrent Neural Networks

Today’s agenda:

1. Review
    - Structure of RNNs
    - Simple RNNs
    - Gated RNNs
2. 1D bounding box revisited

Recurrent Neural Networks

Main applications of RNNs:

- Timeseries or sequence data
  - Weather forecasting, stock market, ...

- Natural language processing
  - Language modeling, translation, ...

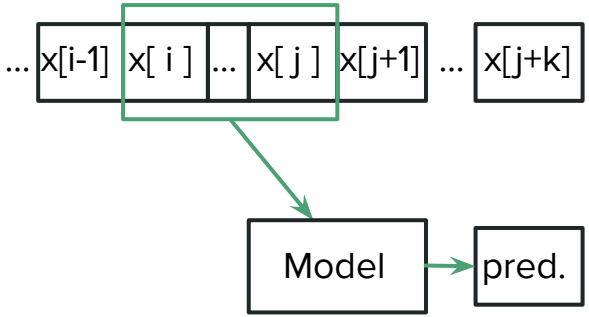
Recurrent Neural Networks

Just as convolutional neural networks account for translations and spatial correlations, RNNs account for

- Repeated patterns
- Causal relations

Repeated patterns

Much like in CNNs, we can account for repeated patterns by utilizing a “sliding window” over the input data.



Causal relations

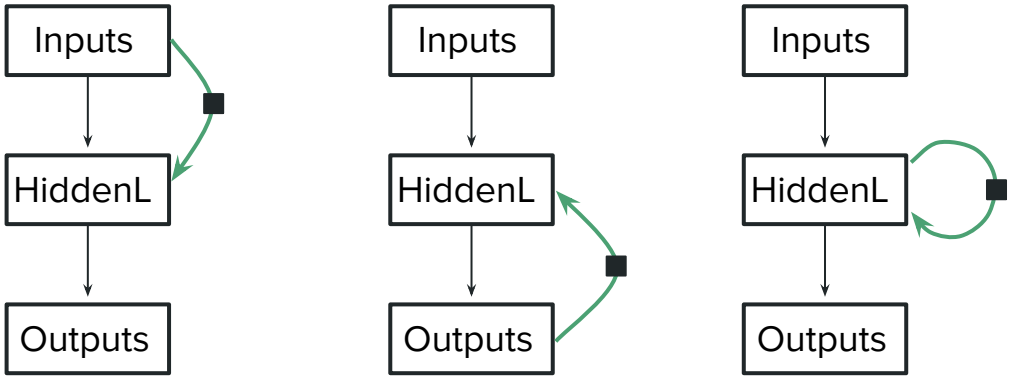
Recurrent neural networks are used to predict the “future” using “past” data, thus they need to have “memory”. This is achieved via a recurrent structure:

$$y[\,i\,] = \text{model}(\, x[i-1], y[i-1]; w, b)$$

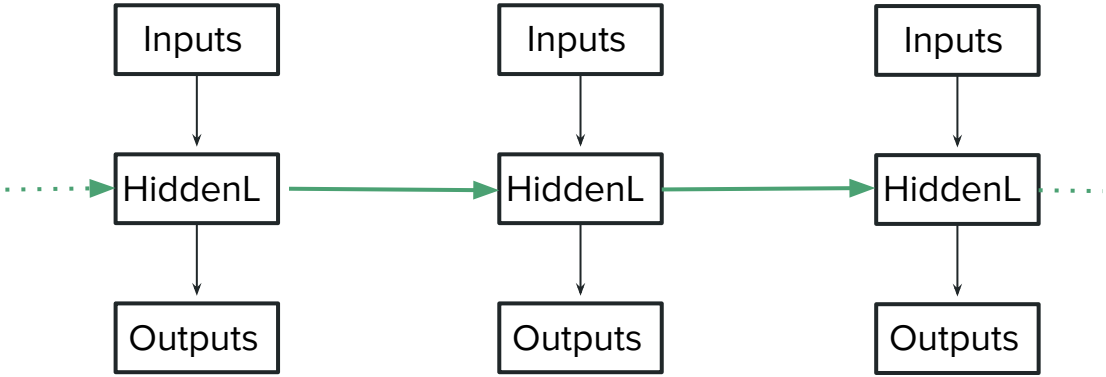
$$y[i+1] = \text{model}(\, x[\,i\,], y[\,i\,]; w, b)$$

$$= \text{model}(\, x[\,i\,], \text{model}(\, x[i-1], y[i-1]; w, b) \,; w, b)$$

Simple RNN


Inputs
HiddenL
Outputs
Inputs
HiddenL
Outputs
Inputs
HiddenL
Outputs

Simple RNN (unfolded)



Gated RNNs

The internal loop inside a recurrent layer adds an extra “depth” to the network. A recurrent layer with long-term memory is similar to a very deep NN and suffers from the same issues.

The good news is that we already have methods to mitigate issues such as vanishing gradient or noise overfitting.

Long short-term memory (LSTM)

One of the methods we employed to improve information propagation in deep NN was adding residual connections. LSTMs basically apply this method to RNNs.

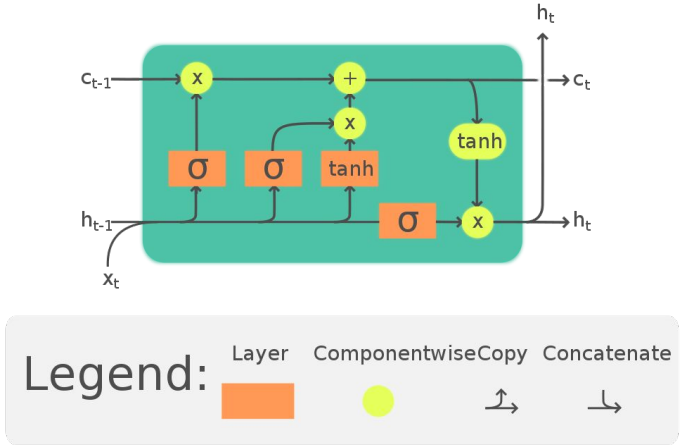


Image from wikipedia

Overfitting

To mitigate overfitting we need to adjust the internal loop of a recurrent layer. We can achieve this in Keras by adding a recurrent_dropout to our RNN:

```
layers.LSTM(10, recurrent_dropout=0.5)
```

Questions?