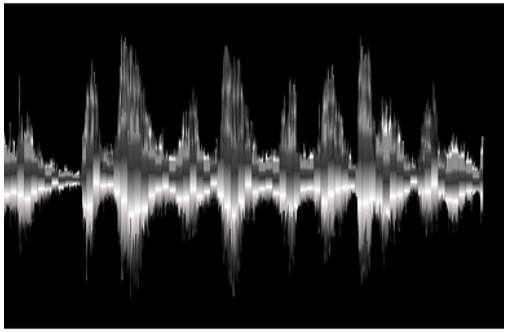
Deep Learning for sequential data

Sequential data

Text, Video, and Audio



Time series: finance, industry, medicine…

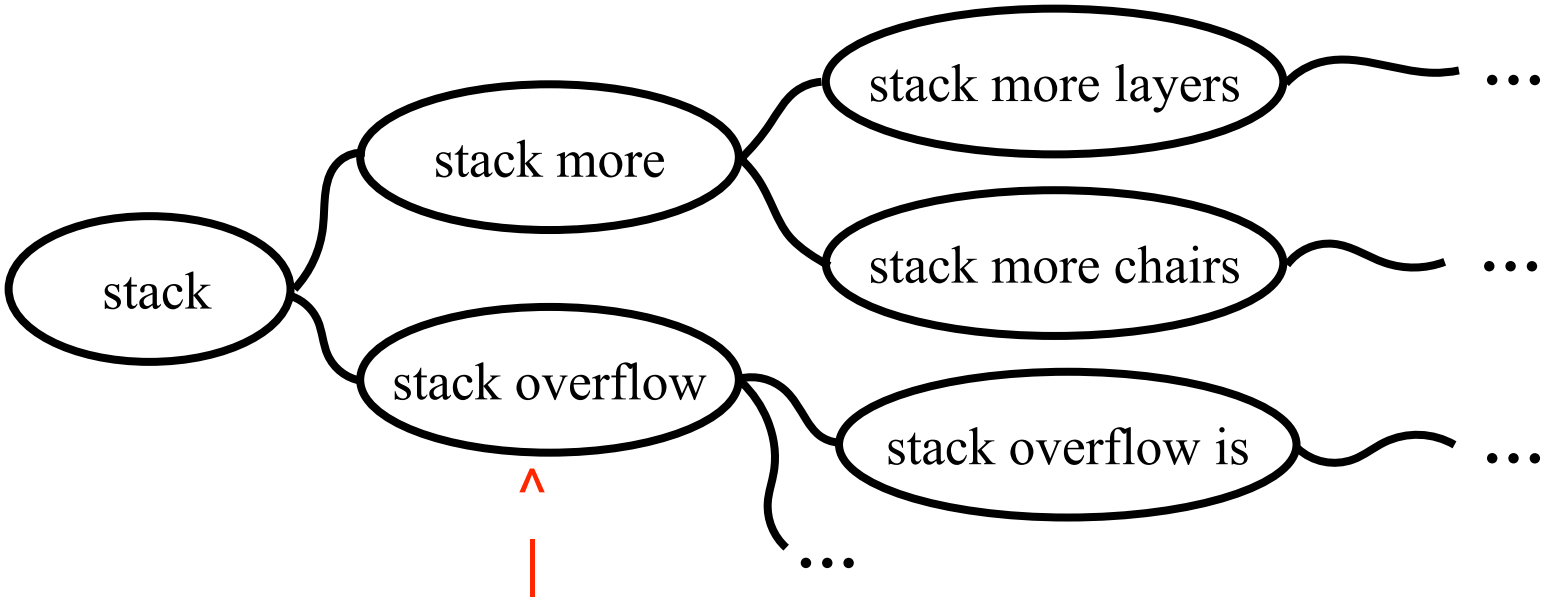
Sequences are everywhere!

Language model

We want to train a generative model of natural language

$$P(text) = P(x_0, \ldots, x_n) =$$
$$= P(x_0)P(x_1|x_0)P(x_2|x_0, x_1)\ldots P(x_n|\ldots)$$

Given some tuple of words x_0, ... x_n, the probability of them appearing simultaneously is the above.



^

|

I think these are the argmax_i

P(i | stack) P(stack)

Language model

We want to train a generative model of natural language

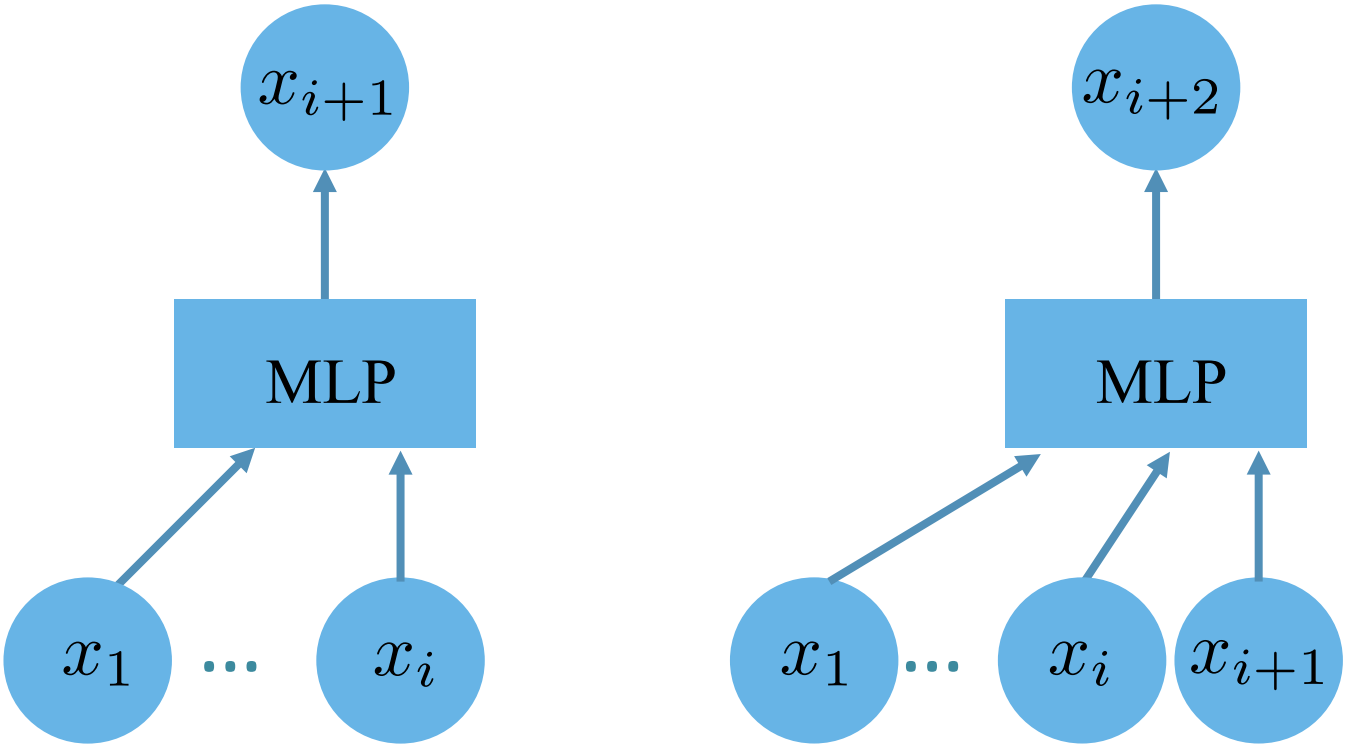
$$P(text) = P(x_0, \ldots, x_n) =$$
$$= P(x_0)P(x_1|x_0)P(x_2|x_0, x_1)\ldots P(x_n|\ldots)$$

Why do we need it?

- Chatbots, question answering
- Machine translation
- Speech recognition
- Any text analysis you can imagine

Why not MLP?

The main problem is arbitrary length of sequences:



i inputs

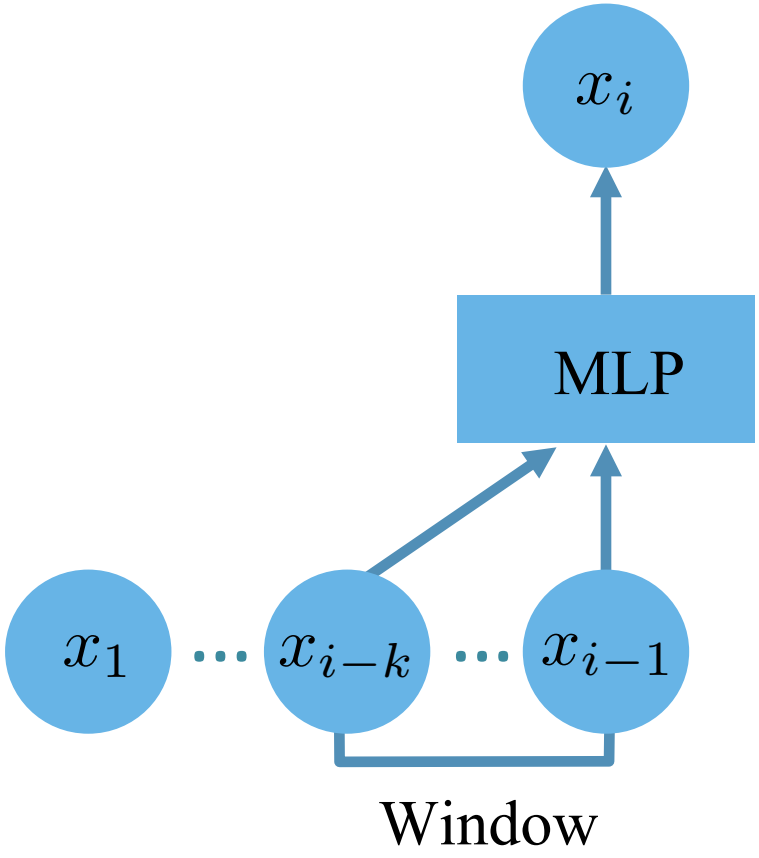
$i+1$ inputs

How can we overcome it?

Why not MLP?

We can use a window of a fixed size as an input.

- This is just a heuristic and it is not clear how to choose the width of the window
- In some tasks we need very wide window therefore there is a problem with the large number of parameters



Why not MLP?

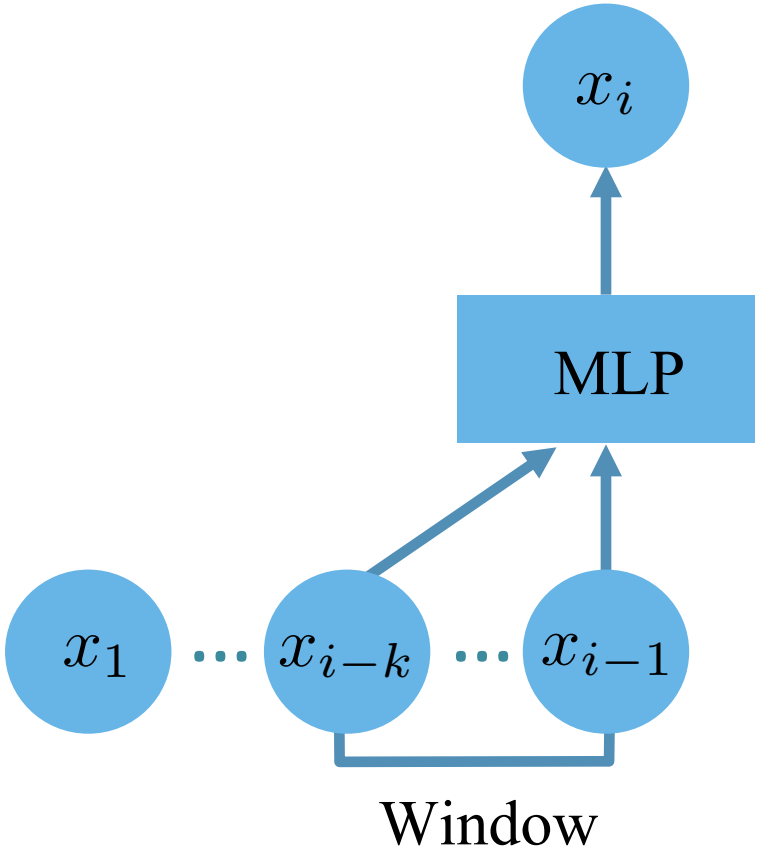
We can use a window of a fixed size as an input.

Question

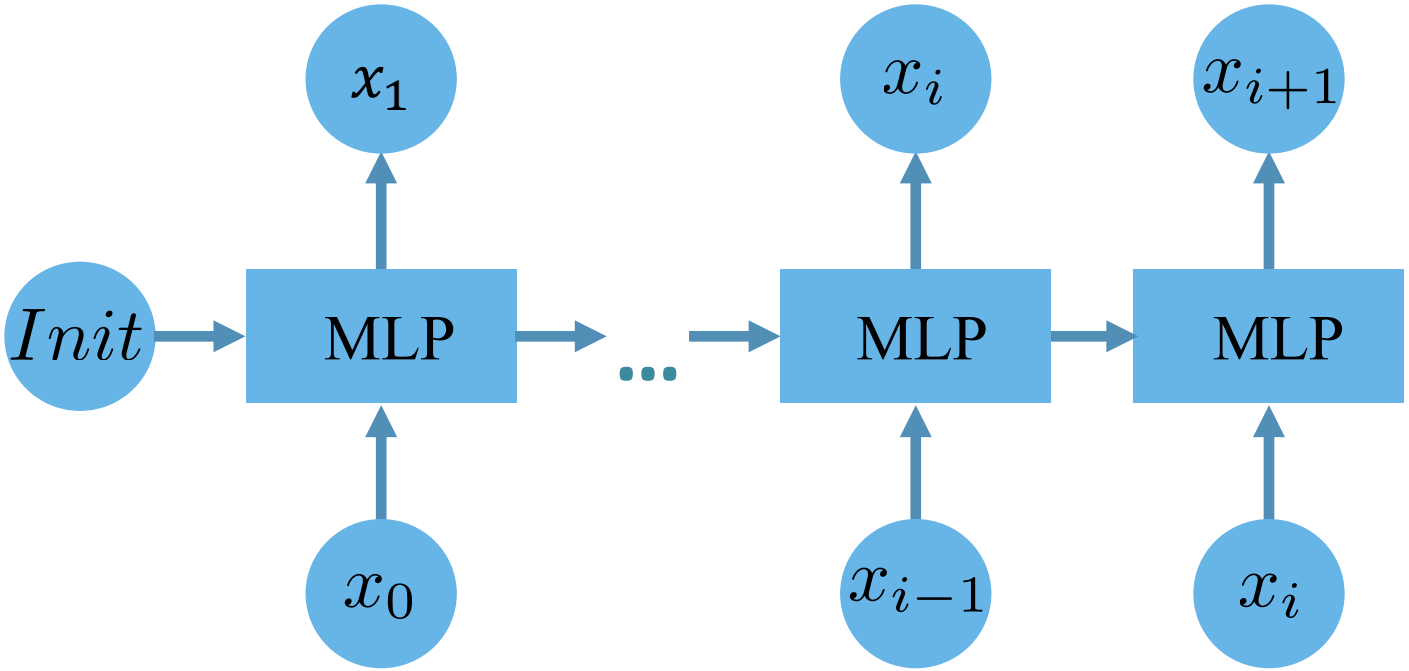
How many weights are there in the first layer of the MLP?

- hidden neurons: 100
- window width: 100
- word embeddings size: 100

More than a million!



Recurrent Architecture



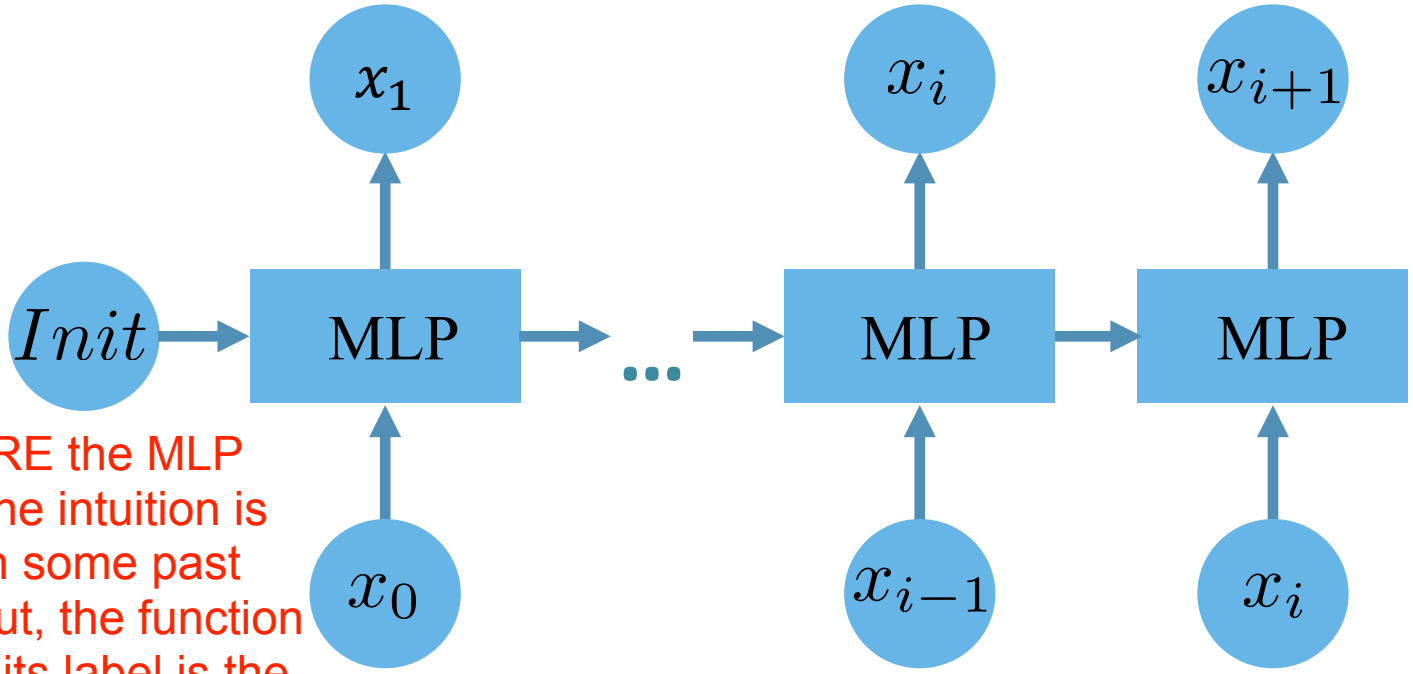
Problem #1: Arbitrary sequence length

Here: Fixed number of inputs at each time step.

At the first step we use some initial vector as an input from previous time step.

per timestep. We see one 'word' on every timestep.

Recurrent Architecture

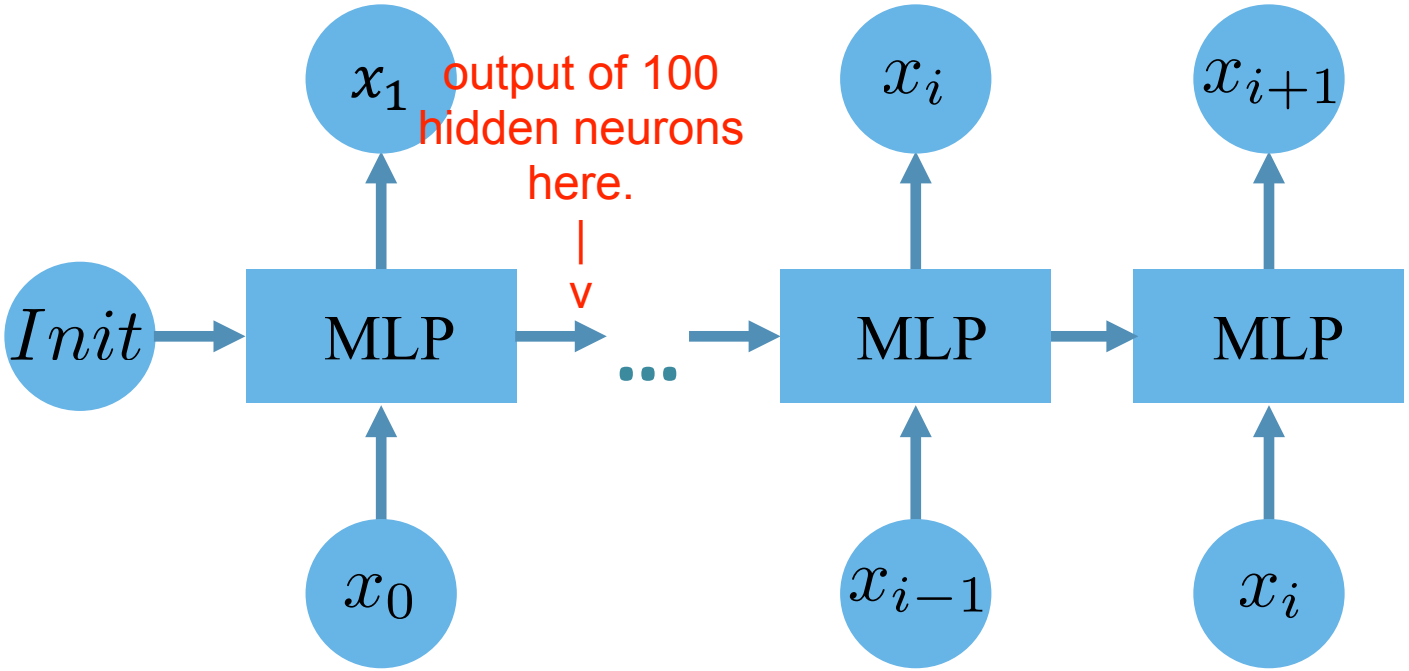


WE SHARE the MLP weights. The intuition is that given some past feature output, the function to compute its label is the same for a given next word

Problem #2: Large number of parameters

Here: All the parameters of an MLP are shared across the different time steps so we need a much smaller number of parameters.

Recurrent Architecture



Question: How many weights are there in the first layer of the MLP?

- hidden neurons: 100
- word embeddings size: 100

There are 200 inputs to the first layer of the MLP: 100 from the current input and 100 from the previous hidden state. With a bias, we have 200 * 100 + 100 = 20100

This is less than

Only 20100!

Summary

- Sequential data is everywhere!
- Feedforward neural network isn't a very natural choice for such data because of arbitrary sequence length and large number of parameters
- Recurrent architecture is much more useful

In the next video:

Simple Recurrent Neural Network:

what is it and how to train it