MACHINE LEARNING

MEI/1

University of Beira Interior, Department of Informatics

Hugo Pedro Proença, hugomcp@di.ubi.pt, 2025/26

Machine Learning

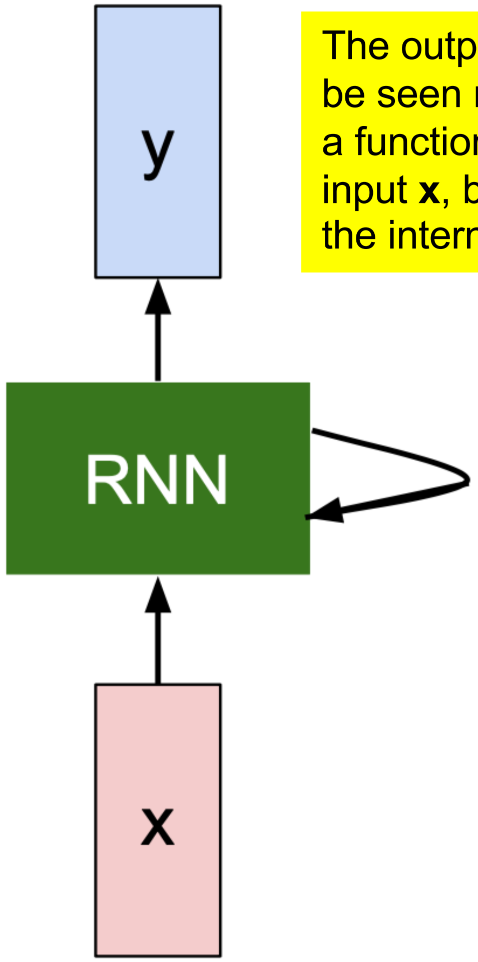
[09]

Syllabus

- Recurrent Neural Networks (RNNs)

Recurrent Neural Networks

- Recurrent Neural Networks (**RNNs**) are deep learning model typically used to process and convert a **sequential data input** into a sequential data **output**.
- **Sequential data**—such as words, sentences, or time-series— have interrelated sequential components, based on complex semantics and syntax rules.
- The key idea in RNNs is to use (apart the classical "weights") an **internal state** that is updated as a sequence is processed



The output **y** can be seen not only as a function of the input **x**, but also of the internal state **h**

Recurrent Neural Networks

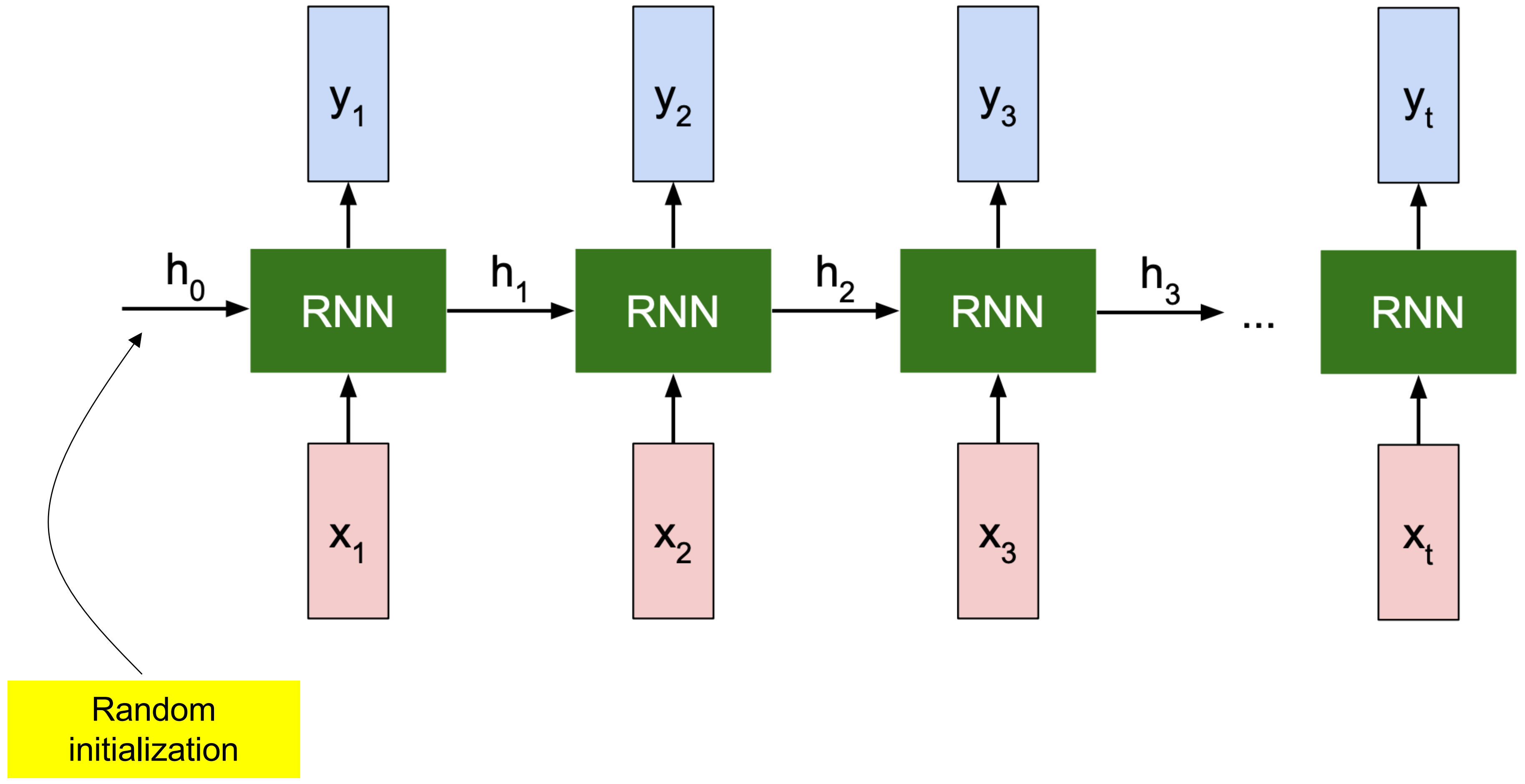
- The forward step of RNNs is divided into two phases:
  - **Step 1:** Obtain the hidden state at time "t" ($\boldsymbol{h}_t$), given the input at time "t" ($\boldsymbol{x}_t$), and the previous state ($\boldsymbol{h}_{t-1}$).

$$\underbrace{\boxed{h_t}}_{\text{new state}} = \underbrace{\boxed{f_W}}_{\substack{\text{some function} \\ \text{with parameters W}}}(\underbrace{\boxed{h_{t-1}}}_{\text{old state}}, \underbrace{\boxed{x_t}}_{\substack{\text{input vector at} \\ \text{some time step}}})$$

  - **Step 2:** Then, obtain the output at time "t" ($\boldsymbol{y}_t$), using the recently updated state ($\boldsymbol{h}_t$).

$$\underbrace{\boxed{y_t}}_{\text{output}} = \underbrace{\boxed{f_{W_{hy}}}}_{\substack{\text{another function} \\ \text{with parameters } W_o}}(\underbrace{\boxed{h_t}}_{\text{new state}})$$

Recurrent Neural Networks



Recurrent Neural Networks

- **Step 1.** To obtain the hidden state at time “t” $(\boldsymbol{h}_t)$, we process a set of inputs $(\boldsymbol{x}_i)$, using the same function $\boldsymbol{f}_W$ at every step.
- In practice, this is due to the fact that backpropagation (weights update) is only done after a batch of steps.

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

- The pioneer architecture (Vanilla RNN) assumes that the state $(\boldsymbol{h}_t)$ is a single hidden vector in the network.
  - “s” is the dimension of the input/output space, and “d” is a hyper-parameter of the RNN.

$$\overset{\text{[d x 1] vector}}{h_t} = \tanh(\underbrace{W_{hh}}_{\text{[d x d] Matrix}} \underbrace{h_{t-1}}_{\text{[d x 1] vector}} + \underbrace{W_{xh}}_{\text{[d x s] Matrix}} \underbrace{x_t}_{\text{[s x 1] vector}})$$

Recurrent Neural Networks

- **Step 2.** Once $\boldsymbol{h}_t$ is found, the output at time "t" $(\boldsymbol{y}_t)$, can also be obtained

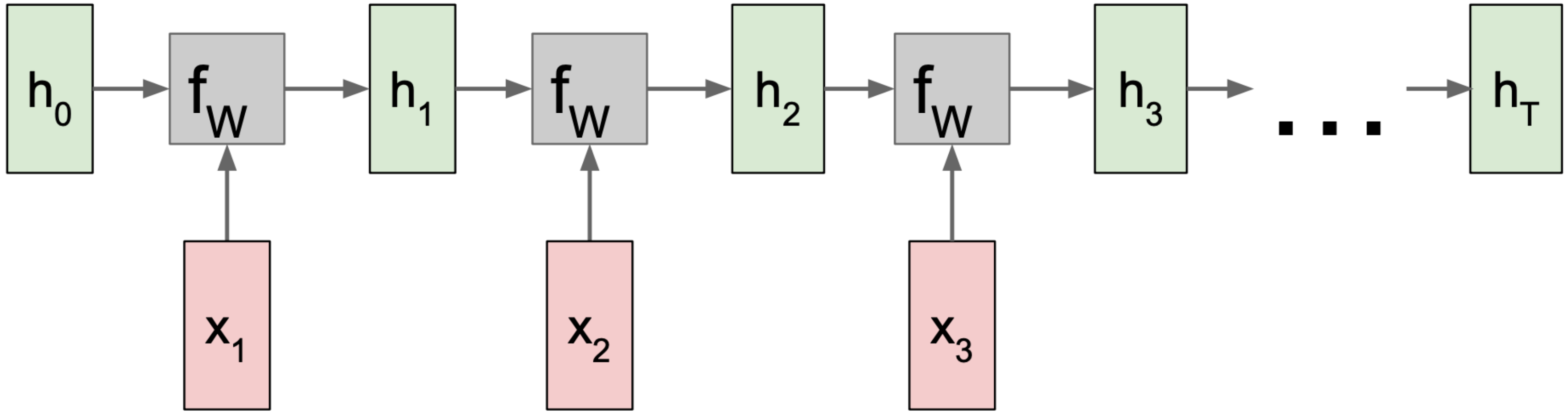
$$y_t = W_{hy} h_t$$

[s x 1] vector

[d x 1] vector

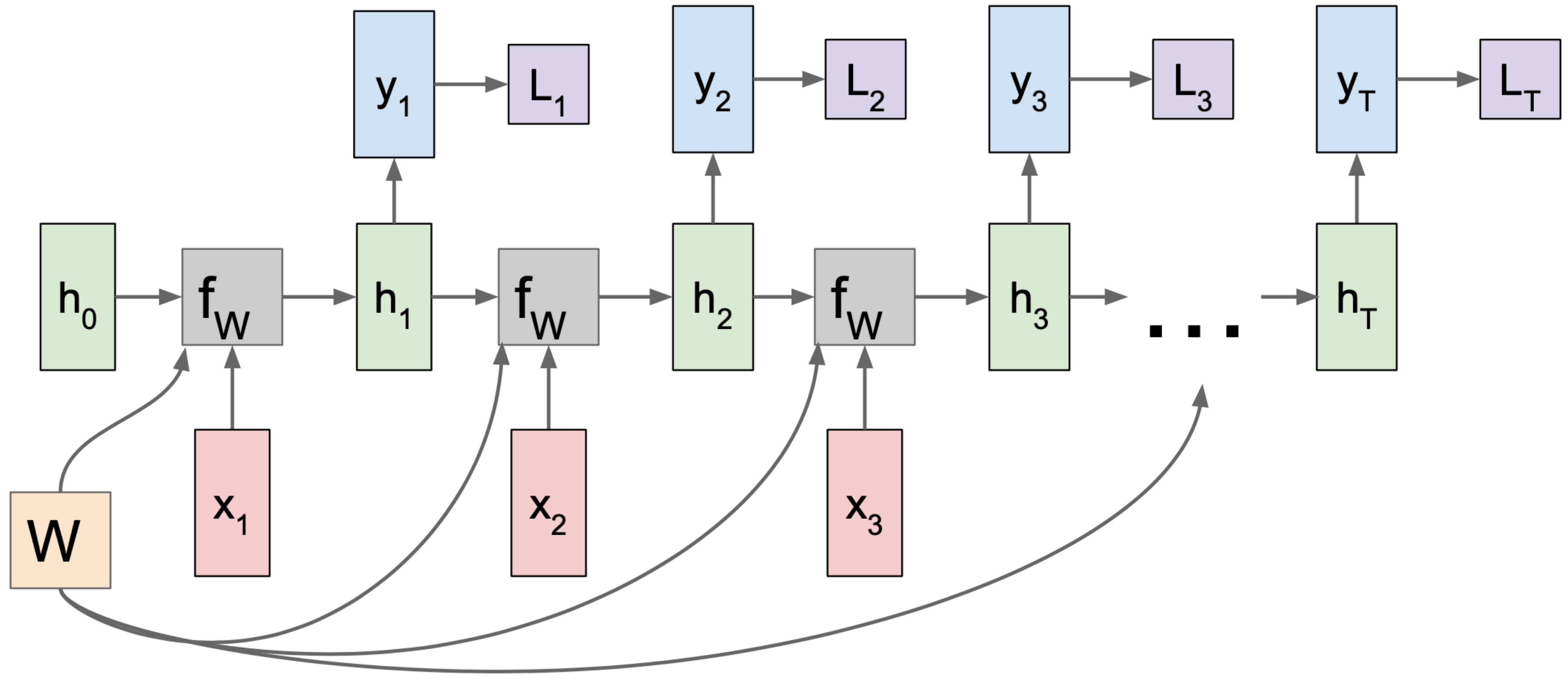
[s x d] Matrix

- Hence, the first step of the corresponding computational graph is given by:



Recurrent Neural Networks

- Only at the second step, the outputs $(\boldsymbol{y}_t)$ are obtained and the partial losses found.
- Such partial loss values are then used to obtain the final loss $\mathcal{L}$ that will be used in backpropagation.



Recurrent Neural Networks: Example

- Text Generation. Consider a single training sequence ("hello").
  - The vocabulary is a set of four symbols: {"h", "e", "l", "o"}
- We start by obtaining a latent representation of each element in the training set. The simplest one is the hot-one encoding.
  - "h" $\rightarrow [1, 0, 0, 0]^T$; "e" $\rightarrow [0, 1, 0, 0]^T$; "$l$" $\rightarrow [0, 0, 1, 0]^T$; "o" $\rightarrow [0, 0, 0, 1]^T$
- More sophisticated content generation techniques (e.g., Chat GPT) obtain richer representations, which elements lie in topological spaces (i.e., neighbor representations are related or are alike).
  - It is reported that these representations play a very important role in the final effectiveness of the model.
- In this example, we are working at the character level. However, "word" or even "small sentence" levels can also be considered.
  - "cat" $\rightarrow [1, 0, \dots, 0, 0]^T$; dog $\rightarrow [0, 1, \dots, 0, 0]^T$;

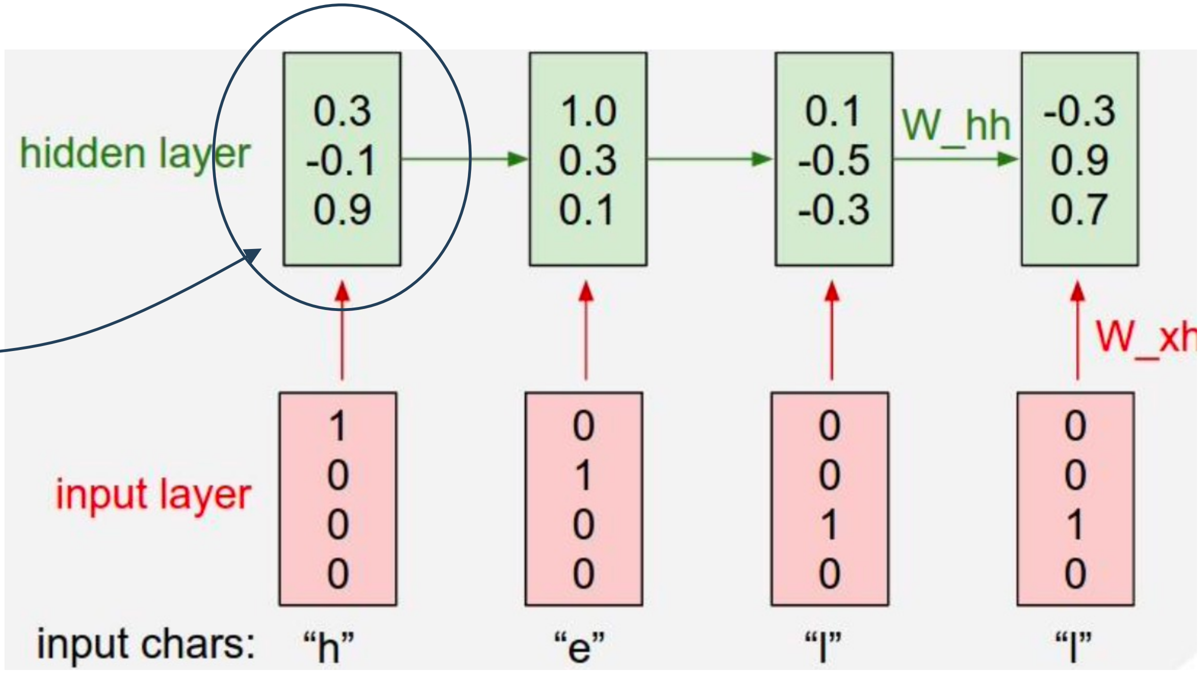
Recurrent Neural Networks: Example

- **Step 1**. Obtain the hidden state representations ($\boldsymbol{h}_t$) for the training sequence ("hell").

Why isn't the complete set considered?

- Suppose that ($\boldsymbol{W}_{hh}$) and ($\boldsymbol{W}_{xh}$) were initialized randomly.

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



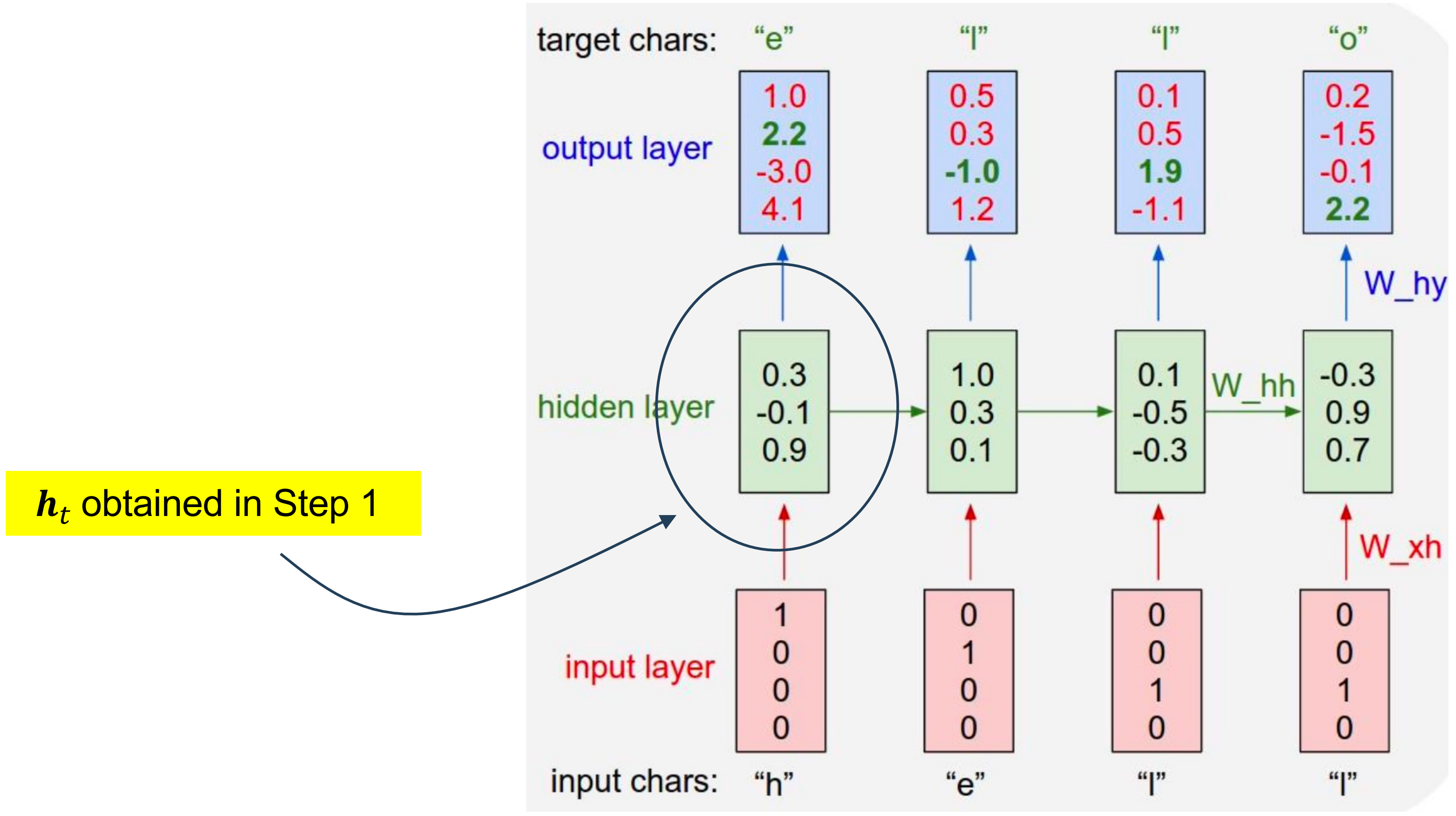
($\boldsymbol{h}_t$)

Recurrent Neural Networks: Example

- **Step 2**. Next, we can obtain the predicted elements at each time.

$$y_t = W_{hy} h_t$$

- Again, suppose that ($\boldsymbol{W}_{hy}$) was initialized randomly.

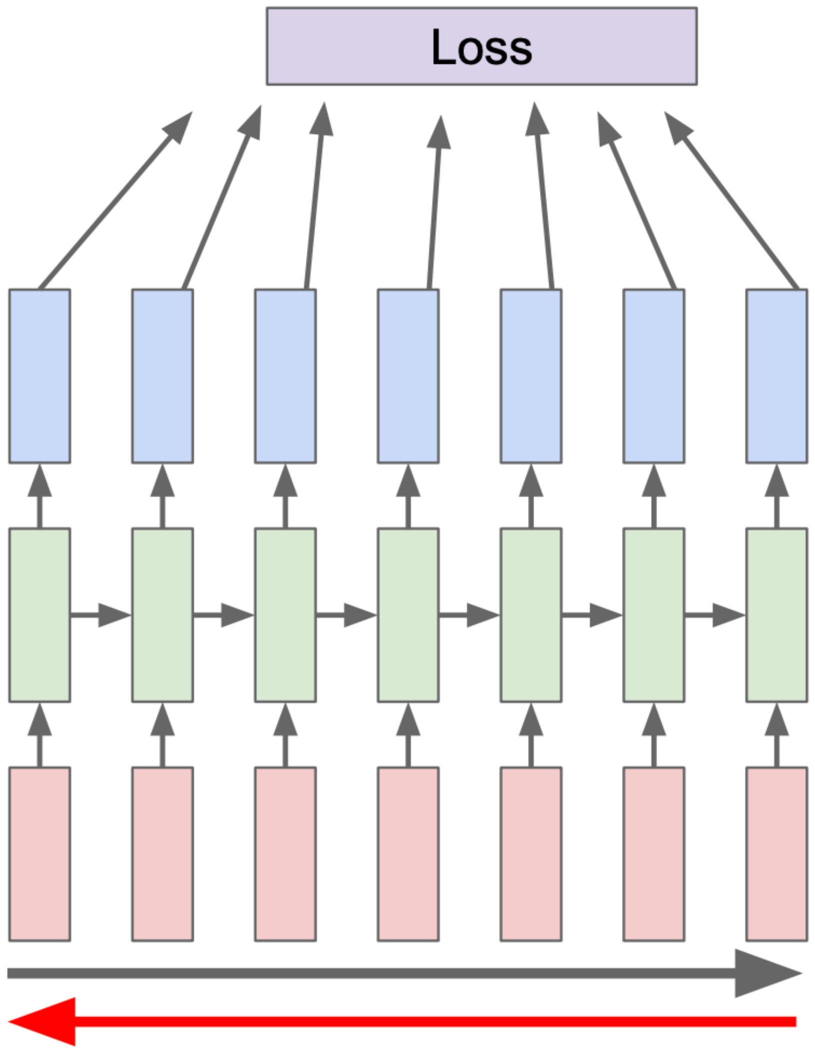


Recurrent Neural Networks: Example

- During training, we forward during the entire sequence to obtain the loss, and then backpropagate to obtain the gradients and adjust the weights.

- However, in practice, we run forward/backward through "**chunks**" instead of the whole sequence.

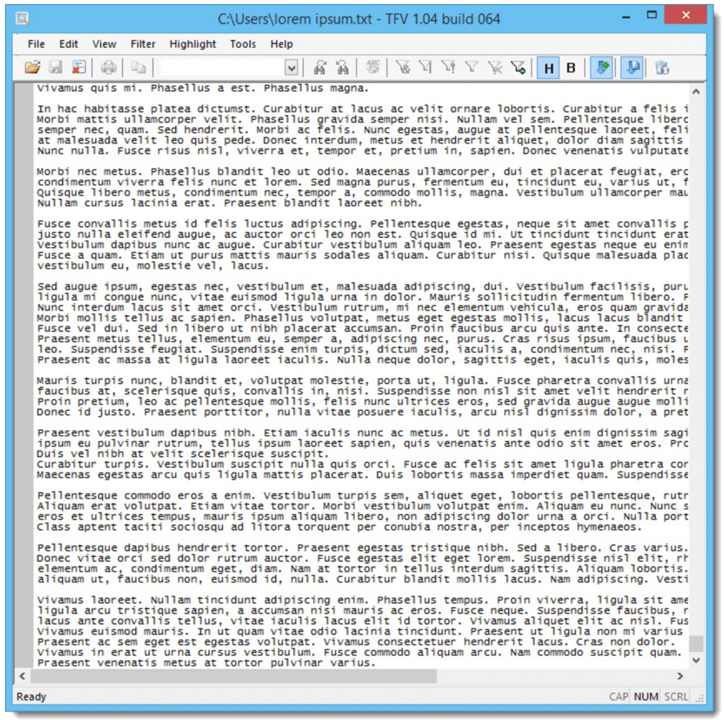
- This is the equivalent to the notion "**batch**" in classical CNNs architectures



Recurrent Neural Networks: Example

- A minimal example (in 112 lines of Python) is available at the web page of this course. It contains a "Vanila" RNN learning process, depending exclusively of "numpy" library. Credits: Andrej Karpathy

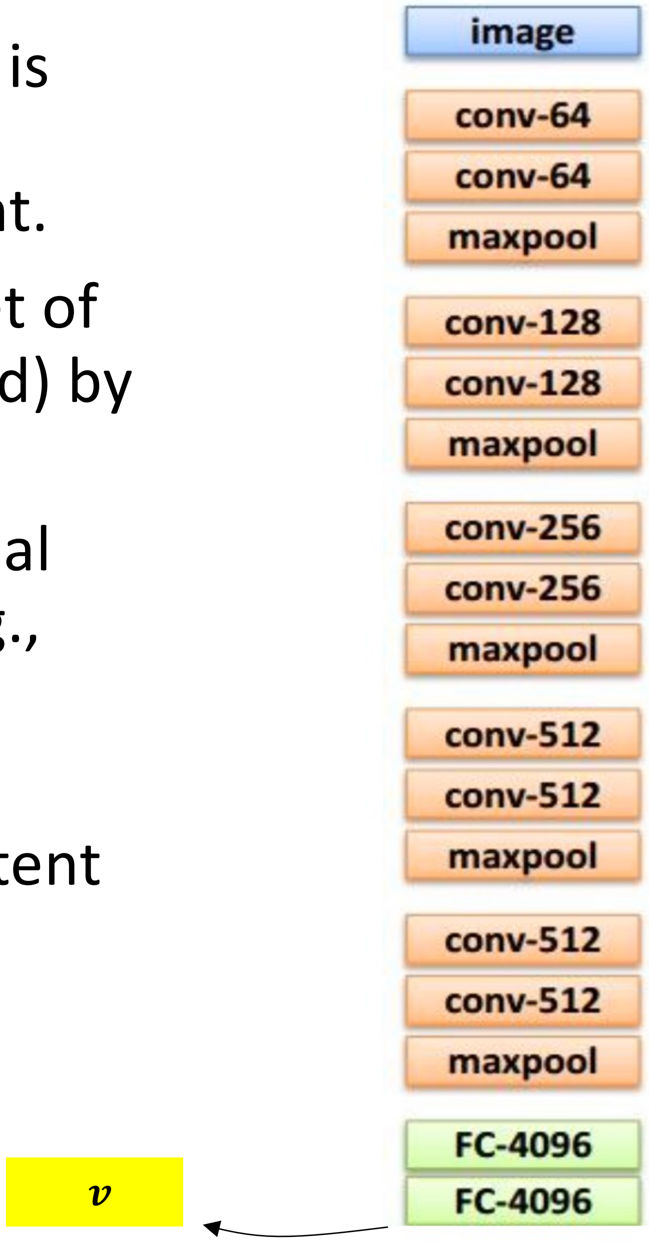
- Based in a simple plain text file (input.txt") it learns to generate text.



```
Vivamus quis mi. Phasellus a est. Phasellus magna.

In hac habitasse platea dictumst. Curabitur at lacus ac velit ornare lobortis. Curabitur a felis i
Morbi mattis ullamcorper velit. Phasellus gravida semper nisi. Nullam vel sem. Pellentesque liberc
semper nec, quam. Sed hendrerit. Morbi ac felis. Nunc egestas, augue at pellentesque laoreet, feli
at malesuada velit leo quis pede. Donec interdum, metus et hendrerit aliquet, dolor diam sagittis
Nunc nulla. Fusce risus nisl, viverra et, tempor et, pretium in, sapien. Donec venenatis vulputate

Morbi nec metus. Phasellus blandit leo ut odio. Maecenas ullamcorper, dui et placerat feugiat, erc
condimentum viverra felis nunc et lorem. Sed magna purus, fermentum eu, tincidunt eu, varius ut, f
Quisque libero metus, condimentum nec, tempor a, commodo mollis, magna. Vestibulum ullamcorper mau
Nullam cursus lacinia erat. Praesent blandit laoreet nibh.

Fusce convallis metus id felis luctus adipiscing. Pellentesque egestas, neque sit amet convallis p
justo nulla eleifend augue, ac auctor orci leo non est. Quisque id mi. Ut tincidunt tincidunt erat
Vestibulum dapibus nunc ac augue. Curabitur vestibulum aliquam leo. Praesent egestas neque eu enim
Fusce a quam. Etiam ut purus mattis mauris sodales aliquam. Curabitur nisi. Quisque malesuada plac
vestibulum eu, molestie vel, lacus.

Sed augue ipsum, egestas nec, vestibulum et, malesuada adipiscing, dui. Vestibulum facilisis, puru
ligula mi congue nunc, vitae euismod ligula urna in dolor. Mauris sollicitudin fermentum libero. F
Nunc interdum lacus sit amet orci. Vestibulum rutrum, mi nec elementum vehicula, eros quam gravida
Morbi mollis tellus ac sapien. Phasellus volutpat, metus eget egestas mollis, lacus lacus blandit
Fusce vel dui. Sed in libero ut nibh placerat accumsan. Proin faucibus arcu quis ante. In consecte
Praesent metus tellus, elementum eu, semper a, adipiscing nec, purus. Cras risus ipsum, faucibus u
leo. Suspendisse feugiat. Suspendisse enim turpis, dictum sed, iaculis a, condimentum nec, nisi. F
Praesent ac massa at ligula laoreet iaculis. Nulla neque dolor, sagittis eget, iaculis quis, moles

Mauris turpis nunc, blandit et, volutpat molestie, porta ut, ligula. Fusce pharetra convallis urna
faucibus at, scelerisque quis, convallis in, nisi. Suspendisse non nisl sit amet velit hendrerit r
Proin pretium, leo ac pellentesque mollis, felis nunc ultrices eros, sed gravida augue augue molli
Donec id justo. Praesent porttitor, nulla vitae posuere iaculis, arcu nisl dignissim dolor, a pret

Praesent vestibulum dapibus nibh. Etiam iaculis nunc ac metus. Ut id nisl quis enim dignissim sagi
ipsum eu pulvinar rutrum, tellus ipsum laoreet sapien, quis venenatis ante odio sit amet eros. Prc
Duis vel nibh at velit scelerisque suscipit.
Curabitur turpis. Vestibulum suscipit nulla quis orci. Fusce ac felis sit amet ligula pharetra cor
Maecenas egestas arcu quis ligula mattis placerat. Duis lobortis massa imperdiet quam. Suspendisse

Pellentesque commodo eros a enim. Vestibulum turpis sem, aliquet eget, lobortis pellentesque, rutr
Aliquam erat volutpat. Etiam vitae tortor. Morbi vestibulum volutpat enim. Aliquam eu nunc. Nunc s
eros et ultrices tempus, mauris ipsum aliquam libero, non adipiscing dolor urna a orci. Nulla port
Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos.

Pellentesque dapibus hendrerit tortor. Praesent egestas tristique nibh. Sed a libero. Cras varius.
Donec vitae orci sed dolor rutrum auctor. Fusce egestas elit eget lorem. Suspendisse nisl elit, rh
elementum ac, condimentum eget, diam. Nam at tortor in tellus interdum sagittis. Aliquam lobortis.
aliquam ut, faucibus non, euismod id, nulla. Curabitur blandit mollis lacus. Nam adipiscing. Vesti

Vivamus laoreet. Nullam tincidunt adipiscing enim. Phasellus tempus. Proin viverra, ligula sit ame
ligula arcu tristique sapien, a accumsan nisi mauris ac eros. Fusce neque. Suspendisse faucibus, r
lacus ante convallis tellus, vitae iaculis lacus elit id tortor. Vivamus aliquet elit ac nisl. Fus
Vivamus euismod mauris. In ut quam vitae odio lacinia tincidunt. Praesent ut ligula non mi varius
Praesent ac sem eget est egestas volutpat. Vivamus consectetuer hendrerit lacus. Cras non dolor.
Vivamus in erat ut urna cursus vestibulum. Fusce commodo aliquam arcu. Nam commodo suscipit quam.
Praesent venenatis metus at tortor pulvinar varius.
```

Recurrent Neural Networks: Applications

- One interesting application of RNNs is "**Image Captioning**", that regards to obtain descriptions for visual content.
- The learning set is composed of a set of images previously labeled (captioned) by humans.
- A classical CNN architecture for global image classification can be used (e.g., VGG or ResNet), removing the final classification layer.
- We use the highest-level possible latent representation



Recurrent Neural Networks: Applications

- The latent representation $\boldsymbol{v}$ is also considered by the RNN, fusing text $\boldsymbol{x}$ to visual information $\boldsymbol{v}$
- A new weights matrix $\boldsymbol{W}_{ih}$ is also required

$$h = \tanh(W_{xh} * x + W_{hh} * h + W_{ih} * v)$$

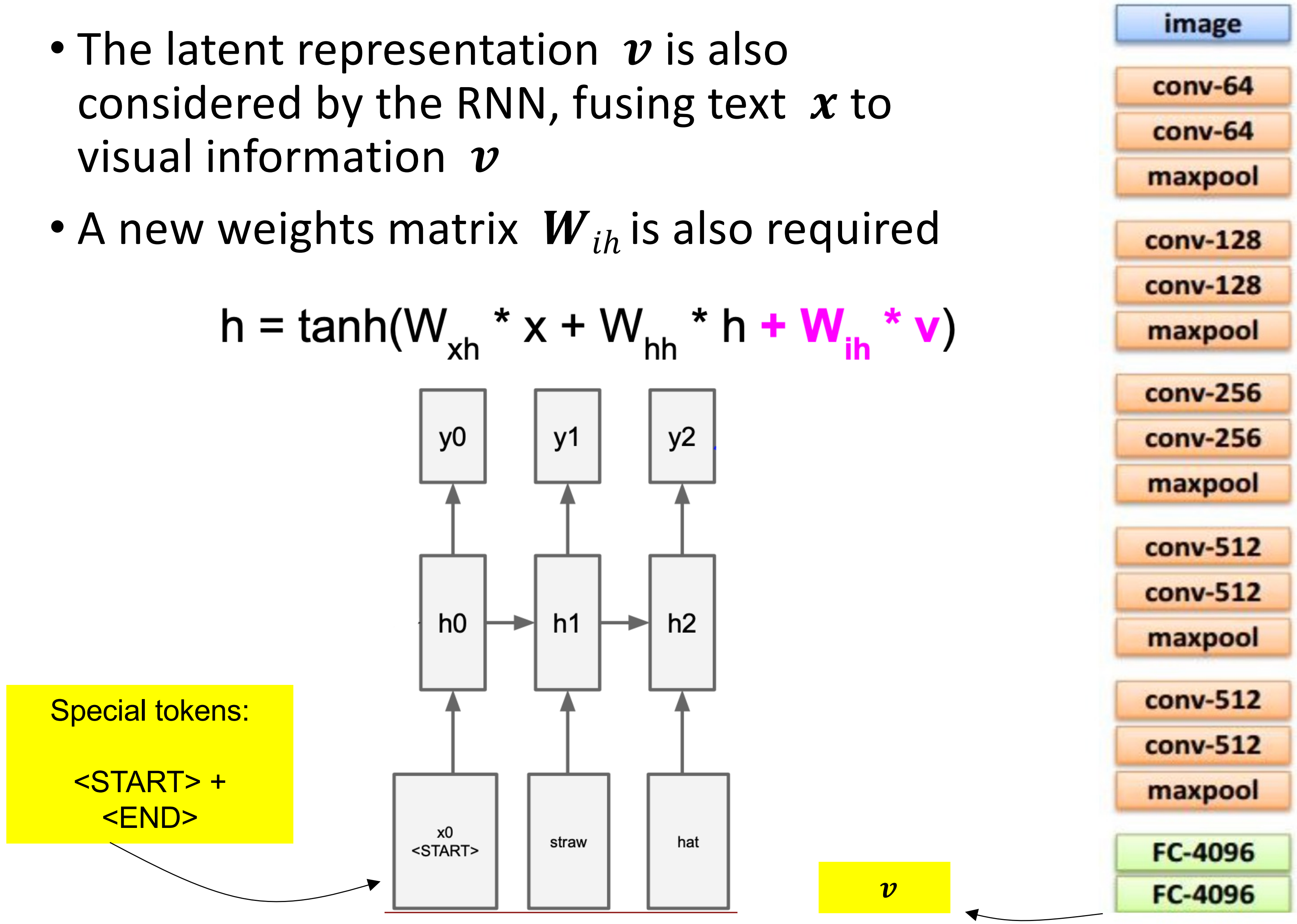
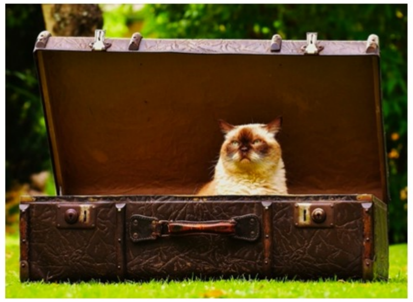
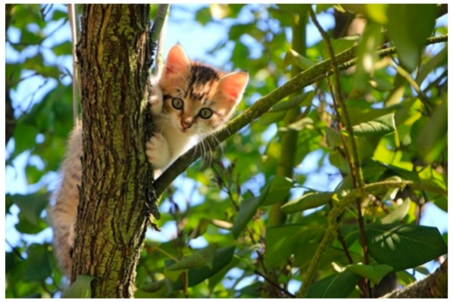
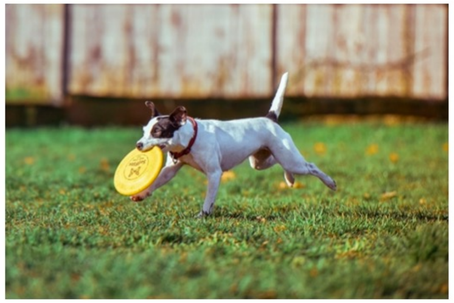
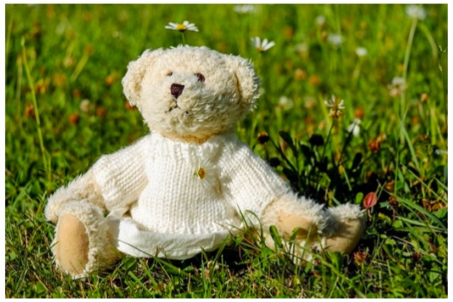


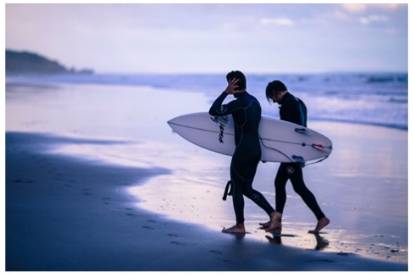
Image Captioning: Results

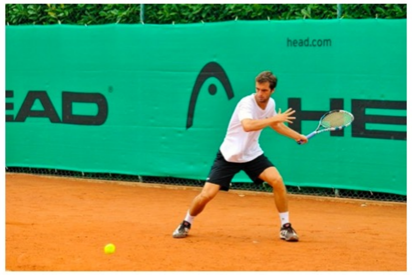

A cat sitting on a suitcase on the floor


A cat is sitting on a tree branch

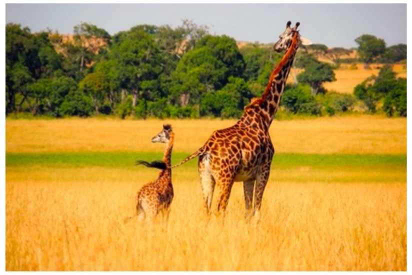

A dog is running in the grass with a frisbee

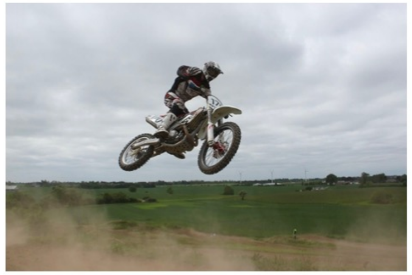

A white teddy bear sitting in the grass


Two people walking on the beach with surfboards



A tennis player in action on the court


Two giraffes standing in a grassy field


A man riding a dirt bike on a dirt track

Credits: Fei-Fei Li, Yunzhu Li, Ruohan Gao