

# REDES RECORRENTES

## LISTA DE EXERCÍCIOS

### Redes Recorrentes

- I) Qual é a principal vantagem das redes neurais recorrentes (RNNs) em relação a MLPs e redes convolucionais?
- II) Descreva a estrutura de uma RNN básica.
- III) Em frameworks de desenvolvimento de redes neurais (como o PyTorch) é comum que o bloco de rede recorrente conte apenas com a camada densa que computa o estado oculto (*hidden state*) e que não conte com a camada densa que computa a saída. Qual motivo disso?
- IV) Uma rede recorrente tem um limite teórico para o tamanho de sequências que consegue processar? E do ponto de vista prático, existem limites?
- V) Redes recorrentes tendem a sofrer mais dos problemas do gradiente que se dissipa e do gradiente explosivo. Por que isso acontece?
- VI) (*Adaptado de CS224N Stanford 2017*) Considere que todos os parâmetros da rede recorrente ilustrada na figura abaixo são escalares ( $\theta_{hh}, \theta_{xh}, b_h, \theta_{hy}$  e  $b_y$ ). A função de ativação  $f[\cdot]$  é a função sinal definida abaixo:

$$f(z) = \begin{cases} z \geq 0, & 1, \\ z < 0, & 0 \end{cases}$$

Sua tarefa é descobrir valores para esses parâmetros que façam a rede recorrente ter como saída inicial o valor 0 e, assim que ela receba um valor 1 como entrada ela passe a ter como saída o valor 1 para toda e qualquer entrada futura. Por exemplo: Se a entrada for 00110 a saída deve ser 00111.

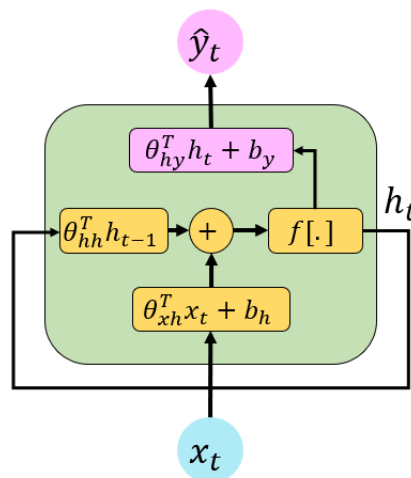


Figure 1: RNN para questão VI

## Gabarito

- I) As RNNs podem capturar e analisar informações sequenciais, tornando-as adequadas para tarefas que envolvem séries temporais, processamento de linguagem natural e dados sequenciais. Elas atingem esse feito usando uma recorrência, ou seja, um estado oculto da rede é computado e serve como entrada para o processamento do próximo item da sequência.
- II) Uma RNN consiste em uma camada de entrada, uma camada oculta com conexões recorrentes e uma camada de saída. A camada oculta mantém um estado oculto que é atualizado a cada passo temporal, e é usado como entrada para os passos subsequentes, permitindo que a rede memorize informações de entradas anteriores.
- III) As tarefas que envolvem dados sequenciais são muito variadas. Podemos fazer análise de sentimentos (classificar uma sequência), *image captioning* no qual temos uma entrada e uma sequência como saída ou ainda *sequence-to-sequence* onde temos sequências na entrada e na saída. Dada essa grande variedade de problemas, seria um desperdício manter parâmetros de saída para cada unidade recorrente. Mantendo a unidade recorrente sem saída por padrão, os *frameworks* transferem a responsabilidade de definir quando a saída deve ser computada (e onde os parâmetros devem ser aprendidos) para o desenvolvedor.
- IV) Do ponto de vista teórico uma RNN consegue processar entradas de qualquer tamanho, um item por vez. Contudo, na prática vemos que RNN não trabalham bem com dependências muito longas. Isso ocorre pois em RNN estamos sempre atualizando o estado oculto em cada uma das iterações ao longo do tempo.
- V) Redes recorrentes são mais sensíveis a esse tipo de problema pois além de serem profundas no número de camadas elas são profundas também na dimensão tempo. Dessa forma, durante o *backpropagation through time* o gradiente sofre mais multiplicações e é acumulado em múltiplas iterações potencializando ambos problemas.
- VI) Existem muitas soluções para essa questão. A solução abaixo é apenas uma das possibilidades:
  - $\theta_{hh} = 1$
  - $\theta_{xh} = 1$
  - $b_h = -1$
  - $\theta_{hy} = 1$
  - $b_y = 0$