Recurrent Neural Networks

Cristian Muñoz

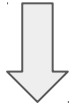
Sumário

1. Lookuptables - Word Embeddings
2. Simple RNN
3. LSTM
4. GRU
5. Bidirecionais
6. Stateful LSTM

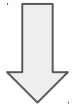
Lookuptables

word2index[“casa”] = 5

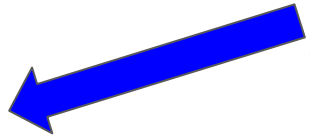
type(Input) = númeric



RNN



type(output) = númeric



index2word[9] = “gato”

Lookuptables

Words Embedding

Wikipedia:

Coleção de nomes de um conjunto de modelos de linguagem e técnicas de aprendizagem de características (features) em NLP. Onde palavras ou frases do vocabulário são mapeados a vetores de números reais.

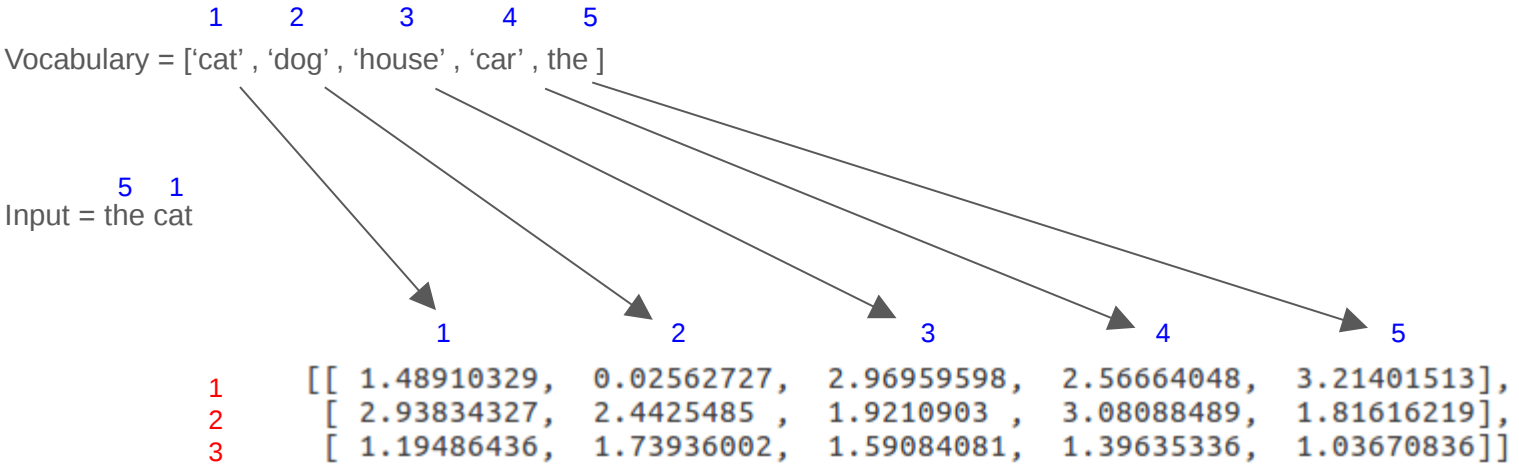
É uma forma de transformar texto em vetores numéricos para logo ser analisados pelos algoritmos de máquinas de aprendizagem que requerem vetores numéricos como entradas.

Forma mais básica: "one-hot-encoding"

Problema com one-hot-encoding: não representa a semelhança entre palavras,

Por exemplo: (cat, dog) , (knife,spoon).

Exemplo:


1 2 3 4 5
Vocabulary = [‘cat’ , ‘dog’ , ‘house’ , ‘car’ , the]
5 1
Input = the cat
1 2 3 4 5
1 [[1.48910329, 0.02562727, 2.96959598, 2.56664048, 3.21401513],
2 [2.93834327, 2.4425485 , 1.9210903 , 3.08088489, 1.81616219],
3 [1.19486436, 1.73936002, 1.59084081, 1.39635336, 1.03670836]]

Exemplo:

1 2 3 4 5

Vocabulary = [‘cat’ , ‘dog’ , ‘house’ , ‘car’ , the]

5 1

Input = the cat

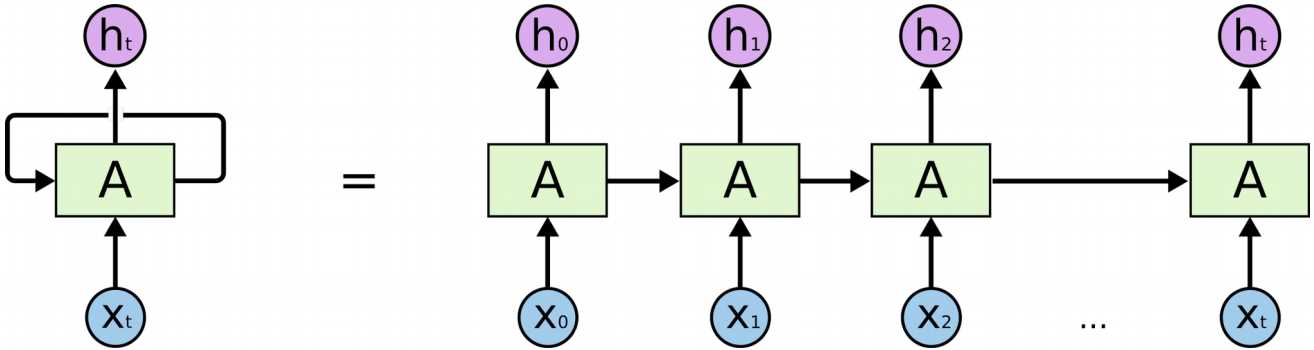
1 2 3 4 5

```
1   [[ 1.48910329,  0.02562727,  2.96959598,  2.56664048,  3.21401513],
2    [ 2.93834327,  2.4425485 ,  1.9210903 ,  3.08088489,  1.81616219],
3    [ 1.19486436,  1.73936002,  1.59084081,  1.39635336,  1.03670836]]
```

Output =

```
[[ 1.48910329, 3.21401513],
 [ 2.93834327, 1.81616219],
 [ 1.19486436, 1.03670836]]
```

Simple RNN



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

Exemplo: Alice in Wonderland

RNN são extensamente usados pela comunidade de Processamento de Linguagem Natural (NLP) em várias aplicações. Por exemplo: **Construção de modelos de linguagem**.

Os modelos de linguagem ajudam a **predecir a probabilidade da seguinte palavra** em função da palavra anterior. Os modelos de linguagem são importantes para várias tarefas de alto nível como **máquinas de tradução**, **correção da pronúncia**, etc.

Dataset: http://www.gutenberg.org/ebooks/11

A ideia nesta prática é a mesma que modelamento baseado em linguagem só que trabalhamos com caracteres em vez de palavras assim a velocidade de processamento é melhor.

Exemplo: Alice in Wonderland - Resultados

==========
Iteration #: 0
Generation from seed: hy do you
hy do you the the the the the the the the the the the the the the the the the the the the the the the the the

==========
Iteration #: 1
Generation from seed: gryphon ha
gryphon har said alice and alice and alice and alice and alice and alice and alice and alice and alice and ali

==========
Iteration #: 2
Generation from seed: ately: hes
ately: hes the the the the the the the the the the the the the the the the the the the the the the the the the

==========
Iteration #: 3
Generation from seed: . he wont
. he wont of the mare the mare the mare the mare the mare the mare the mare the mare the mare the mare the mar

Exemplo: Alice in Wonderland - Resultados

==========
Iteration #: 21
Generation from seed: and the so
and the some of the project gutenberg-tm electronic works to alice said the king, and the one of the this a mi

==========
Iteration #: 22
Generation from seed: et to work
et to work on he was a little that it was a little that it was a little that it was a little that it was a lit

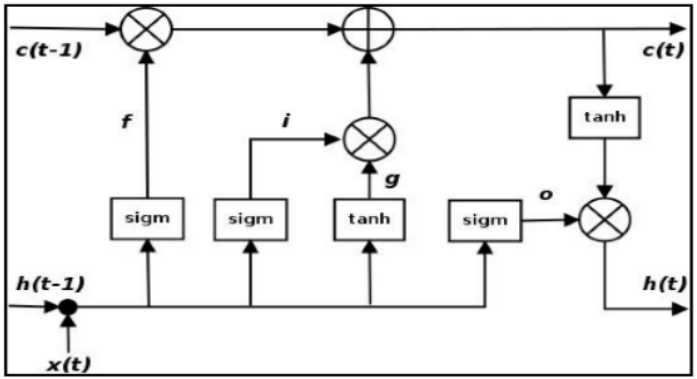
==========
Iteration #: 23
Generation from seed: dry leaves
dry leaves the words with the reasing the did the king said to herself the white rabbit with the reasing the d

==========
Iteration #: 24
Generation from seed: he words h
he words how the mock turtle some the mock turtle some the mock turtle some the mock turtle some the mock turt

LSTM

LSTM é uma variante de RNN que tem a capacidade de aprender dependencias ao longo prazo.

LSTM também implementa recorrência de uma forma similar à **SimpleRNN**, mas em vez de utilizar uma única camada *tanh* existem 4 camadas interseccionando de uma forma bem específica. O seguinte diagrama apresenta a transformações que são aplicadas ao estado oculto no tempo t:

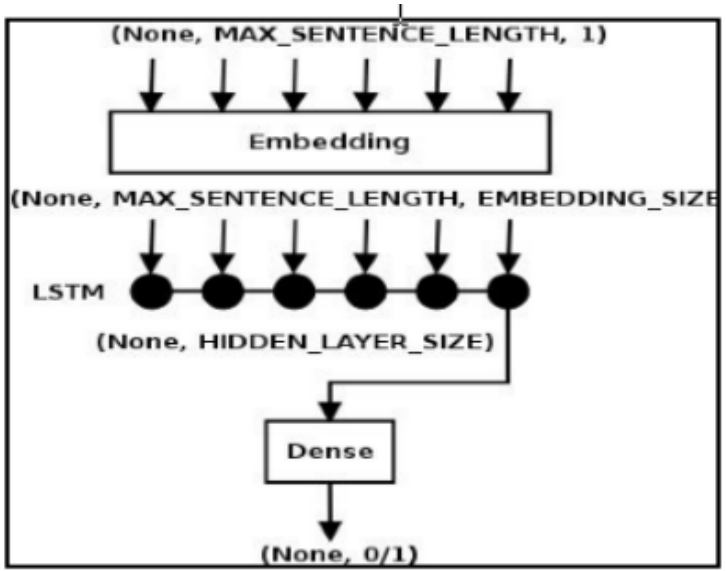


$$
\begin{aligned}
i &= \sigma(W_i h_t + U_i x_t) \\
f &= \sigma(W_f h_{t-1} + U_f x_t) \\
o &= \sigma(W_o h_{t-1} + U_o x_t) \\
g &= tanh(W_g h_{t-1} + U_g x_t) \\
c_t &= (c_{t-1} \otimes f) \otimes (g \otimes i) \\
h_t &= tanh(c_t) \otimes 0
\end{aligned}
$$

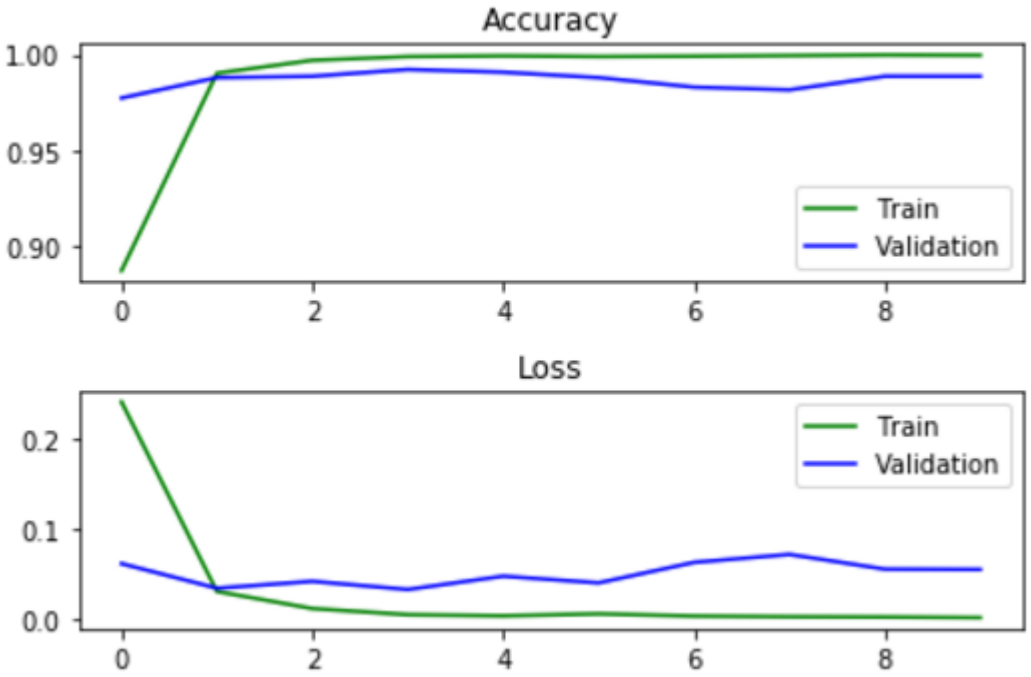
Análise de Sentimentos

Nosso conjunto de treinamento é um dataset de 7000 frases pequenas de UMICH SI650 de uma Competição de Classificação de Sentimentos de Kaggle. Cada frase é rotulada com 1 ou 0 para um sentimento positivo ou negativo respectivamente e nossa RNN aprendera a predecir.

Modelo RRN com LSTM:



Análise de Sentimentos - Resultados

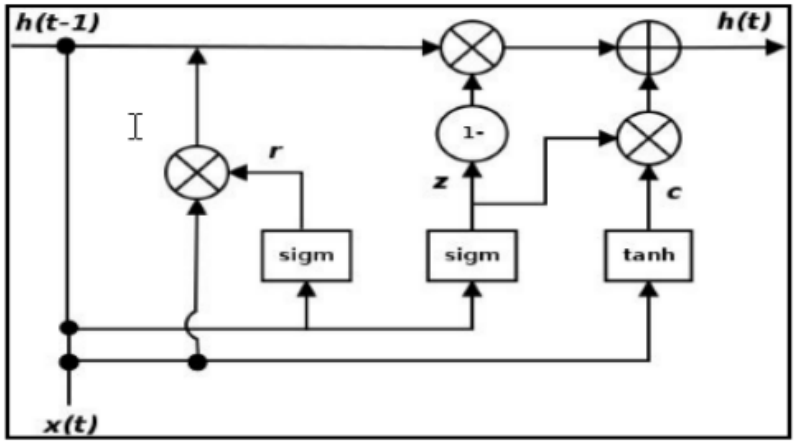


Análise de Sentimentos - Resultados

```
Test score: 0.055, accuracy: 0.989
0        0       harry potter dragged draco malfoy s trousers down past his hips and sucked him into his throat
 with vigor , making whimpering noises and panting and groaning around the blonds rock-hard , aching cock ...
0        0       i hate harry potter , that daniel wotshisface needs a fucking slap ...
0        0       harry potter dragged draco malfoy s trousers down past his hips and sucked him into his throat
 with vigor , making whimpering noises and panting and groaning around the blonds rock-hard , aching cock ...
1        1       harry potter is awesome i do n't care if anyone says differently ! ..
1        1       so as felicia 's mom is cleaning the table , felicia grabs my keys and we dash out like freakin
mission impossible .
```

Gate Recurrent Unit - GRU

GRU é uma variante de LSTM e foi introduzida por K. Cho (Para maior detalhe ler referencia:
Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation
, K. Cho, 2014). Ele consegue manter a resistência do LSTM ao problema do desaparecimento da gradiente, mas sua estrutura interna é simple, pelo qual é mais rápido para treinar, pois poucos cálculos são necessários pa



$$z = \sigma(W_z h_{t-1} + U_z x_t)$$
$$r = \sigma(W_r h_{t-1} + U_r x_t)$$
$$c = tanh(W_c(h_{t-1} \otimes r) + U_c x_t)$$
$$h_t = (z \otimes c) \oplus ((1 - z) \otimes h_{t-1})$$

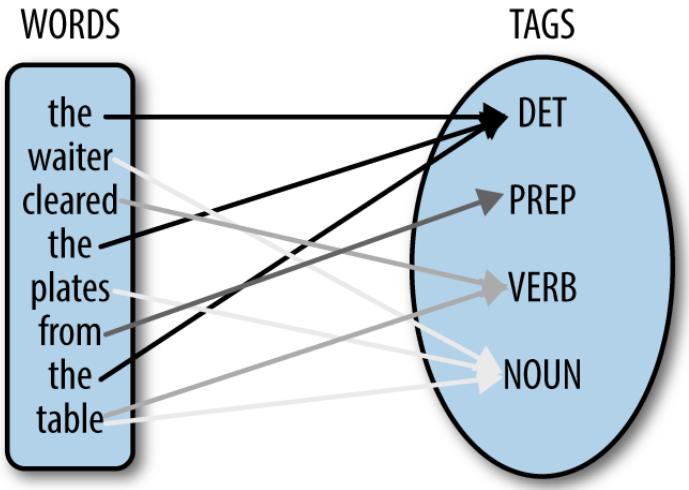
z : Porta de atualização

r : Porta de reset

POS (Part of speech) tagging

POS tagging é a categorização gramatical de palavras que são usadas da mesma forma através de múltiplos frases. Exemplo de POS tagging: nomes, verbos, adjetivos, etc.

POS tagging normalmente é realizado manualmente, mas atualmente é feito automaticamente utilizando modelos estatísticos. Nos últimos anos Deep Learning tem sido aplicado nestes problemas com bons resultados(para maior informação ler artigo: Natural Language Processing (almost) from Scratch , R. Collobert, Journal of Machine Learning Research. 2011).



Exemplo : POS tagging

Para nossos dados de treinamento, precisamos de frases marcadas com parte das tags da fala. O dataset The Peen Treebank, é uma anotação de 4.5 milhões de palavras de Inglês Americano. A data não é livre, mas um 10% do dataset é disponivel em NLTK (http://www.nltk.org/), que utilizaremos em nossa rede.

Aplicações: Named Entity Resolution, Coreference Resolution, Sentiment Analysis and Question Answering.

https://cs.nyu.edu/grishman/jet/guide/PennPOS.html

Alphabetical list of part-of-speech tags used in the Penn Treebank Project:

Number	Tag	Description
1.	CC	Coordinating conjunction
2.	CD	Cardinal number
3.	DT	Determiner
4.	EX	Existential *there*
5.	FW	Foreign word
6.	IN	Preposition or subordinating conjunction
7.	JJ	Adjective
8.	JJR	Adjective, comparative
9.	JJS	Adjective, superlative
10.	LS	List item marker
11.	MD	Modal
12.	NN	Noun, singular or mass
13.	NNS	Noun, plural
14.	NNP	Proper noun, singular
15.	NNPS	Proper noun, plural
16.	PDT	Predeterminer
17.	POS	Possessive ending
18.	PRP	Personal pronoun
19.	PRP$	Possessive pronoun
20.	RB	Adverb
21.	RBR	Adverb, comparative
22.	RBS	Adverb, superlative
23.	RP	Particle
24.	SYM	Symbol
25.	TO	*to*
26.	UH	Interjection
27.	VB	Verb, base form
28.	VBD	Verb, past tense
29.	VBG	Verb, gerund or present participle
30.	VBN	Verb, past participle
31.	VBP	Verb, non-3rd person singular present
32.	VBZ	Verb, 3rd person singular present
33.	WDT	Wh-determiner
34.	WP	Wh-pronoun
35.	WP$	Possessive wh-pronoun
36.	WRB	Wh-adverb

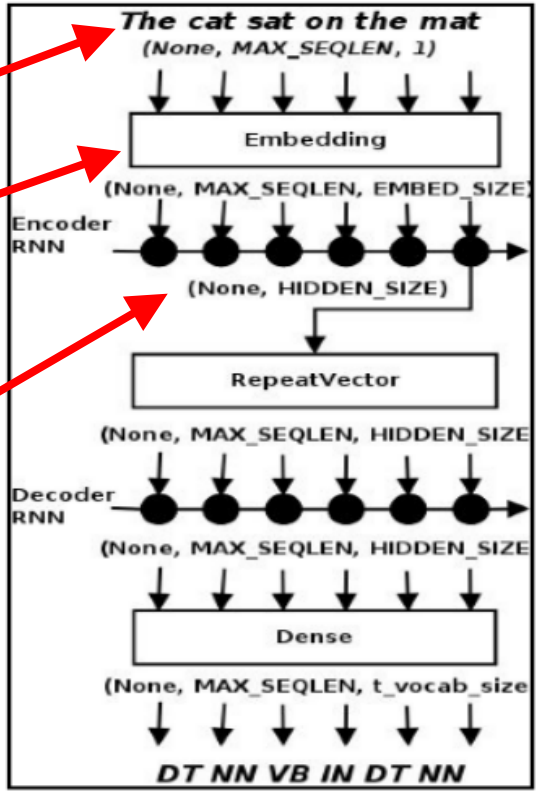
Exemplo : POS tagging

Batch_size: Indeterminado (None)

Entrada: Tensor de índices de palavras de dimensão: **(None, MAX_SEQLEN, 1)**.

Embedding: Converte cada palavra num vetor da forma (EMBED_SIZE), pelo qual o tensor de saída desta camada tem a dimensão **(None, MAX_SEQLEN, EMBED_SIZE)**.

Encoder GRU: sai com uma dimensão HIDDEN_SIZE. A camada GRU é configurada para retornar um único vetor de contexto (**return_sequences=False**) depois de observar uma sequencia de dimensão MAX_SEQLEN, então o tensor de saída da camada GRU tem a forma **(None, HIDDEN_SIZE)**.



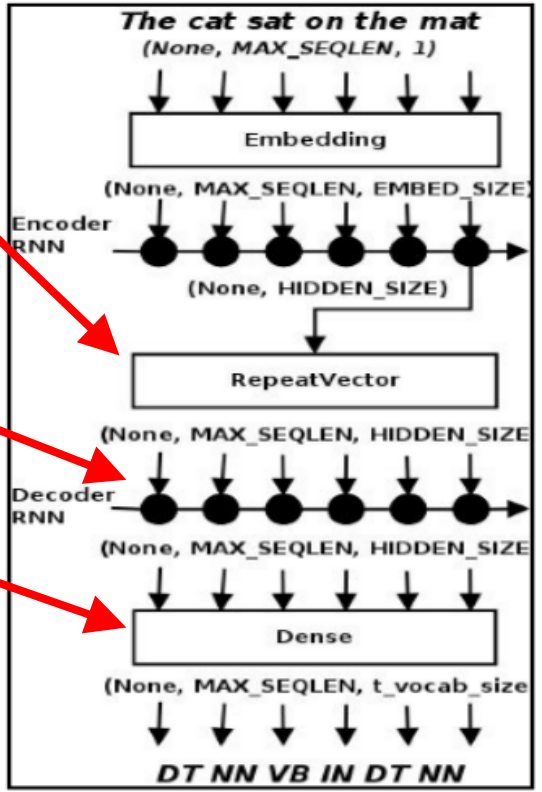
Exemplo : POS tagging

RepeatVector: O vetor de contexto é logo replicado em um tensor da forma **(None, MAX_SEQLEN, HIDDEN_SIZE)**.

Decoder GRU: (**return_sequences=True**): Saida: **(None, MAX_SEQLEN, HIDDEN_SIZE)**.

Camada FC: que produce um vetor de saída de dimensão **(None, MAX_SEQLEN, t_vocab_size)**.

Função de ativação: na camada FC é softmax. O argumento máximo (argmax) em cada coluna do tensor é indexada para predecir o POS tag para a palavra em essa posição.



Exemplo : POS tagging

Trocar as camadas **GRU** por **LSTM** ou **SimpleRNN.**

Modelo Bidirecional

Em algumas aplicações é completamente possível que a saída seja também dependente de uma saída futura.

NLP: Os atributos de uma palavra ou frases que tentamos predecir podem depender do contexto dada pela frase completa, não só das palavras que precedem ela.

RNN bidirecionais também ajuda a arquitetura da rede ao dar igual importância ao início e final de a sequência, e incrementa os dados disponíveis para treinamento.

RNN bidirecionais são 2 RNN acoplados uma acima de outra, lendo a entrada no sentido oposto.

Treinar o problema anterior com uma RNN Bidireccional

Stateful LSTM

RNN podem manter seus estados ocultos ao longo dos batches no treinamento. O estado oculto calculado para o batch dos dados de treinamento serão utilizados para inicializar os estados ocultos para o seguinte batch.

Beneficio:

Menor dimensão e/ou menor tempo de treinamento.

Desvantagem:

O cientista é responsável por treinar a rede com uma dimensão do batch dependente da periodicidade dos dados, e reiniciar os dados após cada época.

Também:

Os dados não podem embaralhados para a uniformidade, pois o ordem dos dados é relevante para o estado da rede.

Exemplo: Previsão do consumo de eletricidade

Dataset:

Electricity Load Diagrams de UCI Machine Learning Repository.

Contém:

Informação de consumo de 370 clientes, com dados tomados em intervalos de 15 minutos em um período de 4 anos desde 2011 até 2014.

Foi selecionado o cliente # 250 para nosso exemplo.

Treinar o modelo

FIM