法律声明

- 本课件包括：演示文稿，示例，代码，题库，视频和声音等，小象学院拥有完全知识产权的权利；只限于善意学习者在本课程使用，不得在课程范围外向任何第三方散播。任何其他人或机构不得盗版、复制、仿造其中的创意，我们将保留一切通过法律手段追究违反者的权利。

LessonTwo:自然语言处理

主　讲：赵英俊

课时大纲

- 自然语言处理理论基础
- 用到的高级API详解
- 项目和代码设计思路
- 手把手开始编程

自然语言处理理论基础

计算 $P(w_1, w_2, ..., w_n)$

马尔可夫(Markov)假设：

“无记忆性”：未来的事件，只取决于有限的历史

$$P(w_5 | w_4, w_3, w_2, w_1)$$

$P(w_5)$ unigram

$P(w_5 | w_4)$ bigram

$P(w_5 | w_4, w_3)$ trigram

$$P(w_1, w_2, ..., w_n) = P(w_1)P(w_2 | w_1)...P(w_n | w_{n-1})$$

$$P(w_1, w_2, ..., w_n) = P(w_1 | START)P(w_2 | w_1)...P(w_n | w_{n-1})P(EOS | w_n)$$

自然语言处理理论基础

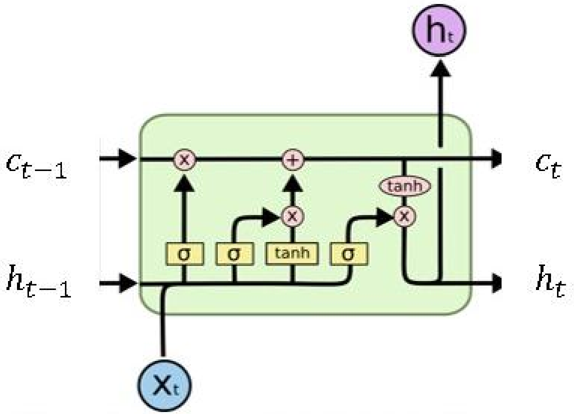
- 优化目标函数
  - Batch Gradient Descent
    - 必须遍历所有的训练数据才更新一次
  - Stochastic Gradient Descent
    - 每看见一个数据点就更新，非常不稳定
  - Mini-batch Gradient Descent Mini-batch梯度下降法
    1. 初始化参数
       - $\Theta = \text{uniform}(d)$
    2. 随机抽取m个数据点$T_m = \{(x_i, y_i) | i = k_1, \ldots, k_m\}$,计算偏导数
       - $\nabla\Theta = \frac{\partial \text{Obj}(\Theta, T_m)}{\partial\Theta} = \frac{\partial}{\partial\Theta} \sum_{i=k_1}^{k_m} Obj(\Theta, x_i, y_i)$
    3. 更新参数
       - $\Theta = \Theta - \eta \nabla\Theta$
    4. 适当的条件更新learning rate η，返回2，直到收敛

自然语言处理理论基础

- Recurrent Neural Network 循环神经网络
  - $\frac{\partial J}{\partial W_t} = \frac{\partial J}{\partial h_{t+n}} \frac{\partial h_{t+n}}{\partial h_{t+n-1}} \cdots \frac{\partial h_{t+1}}{\partial h_t} \frac{\partial h_t}{\partial W_t}$
  - $\frac{\partial h_{t+n}}{\partial h_{t+n-1}} \cdots \frac{\partial h_{t+1}}{\partial h_t} = \prod_{i=t}^{t+n-1} \frac{\partial \tanh(a_i)}{\partial a_i} W$
  - 假设W是一个一维的矩阵：
    - $|W| < 1,\ \prod_{i=t}^{t+n-1} W \to 0$
      - Vanishing Gradients (梯度消失)
    - $|W| > 1,\ \prod_{i=t}^{t+n-1} W \to \infty$
      - Exploding Gradients (梯度爆炸)
  - 如果W是高维的矩阵呢?
    - $|W| = \lambda_{max}(W)$, 最大的特征值

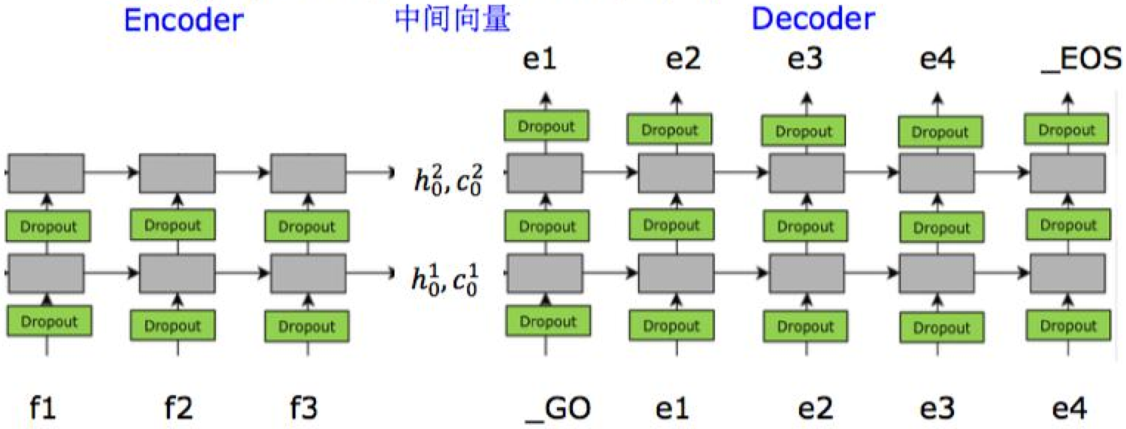
自然语言处理理论基础

- Long Short-Term Memory (LSTM)
  - 尝试解决的问题：Vanishing Gradients
    - Vanilla RNN: $h_t = f(h_{t-1}, x_t)$
    - LSTM: $h_t, c_t = f(h_{t-1}, c_{t-1}, x_t)$



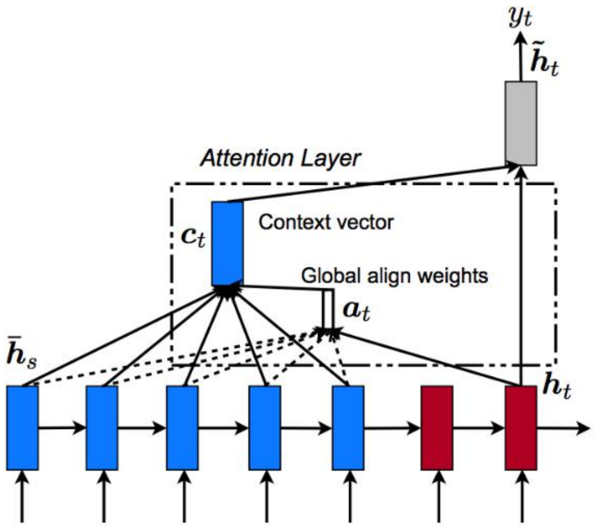
自然语言处理理论基础

- seq2seq的结构
  - 核心：$p(e_1^N|f_1^M) = \prod_{i=1}^{N} p(e_i|e_1^{i-1}, f_1^M)$
  - h_0, c_0由另外一个LSTM来计算

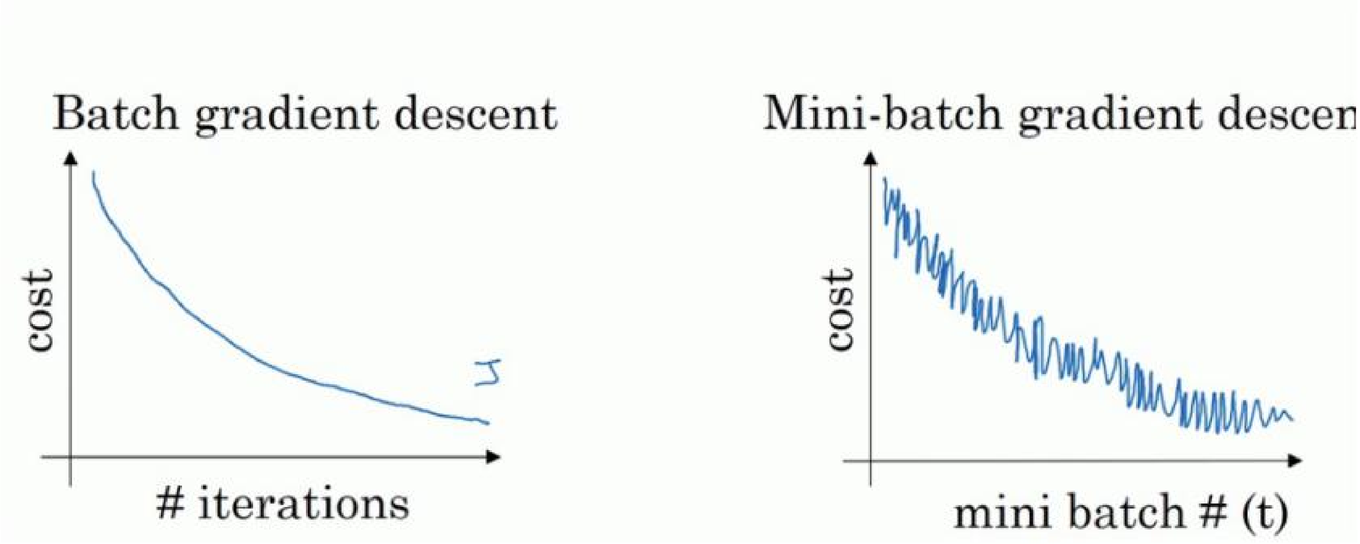


自然语言处理理论基础

- Attention

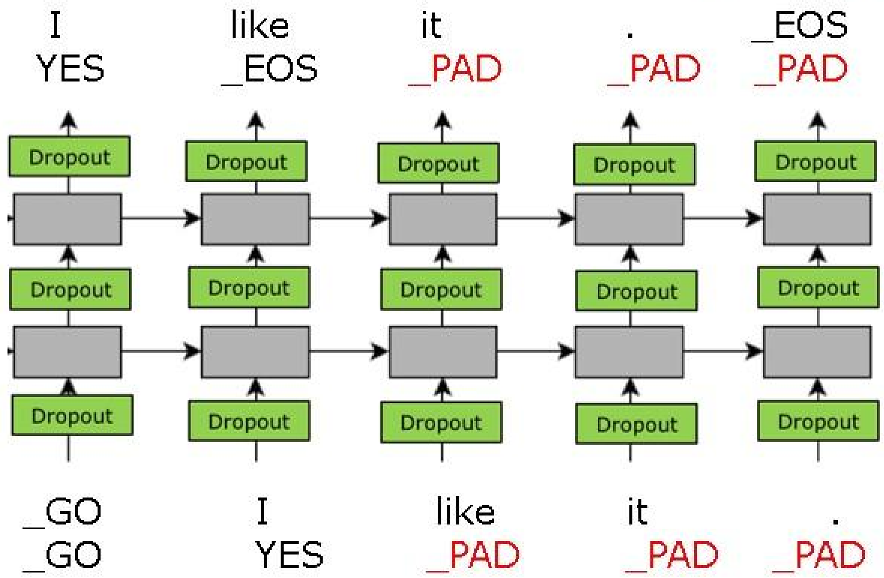


自然语言处理理论基础


Batch gradient descent
cost
J
iterations
Mini-batch gradient descen
cost
mini batch # (t)

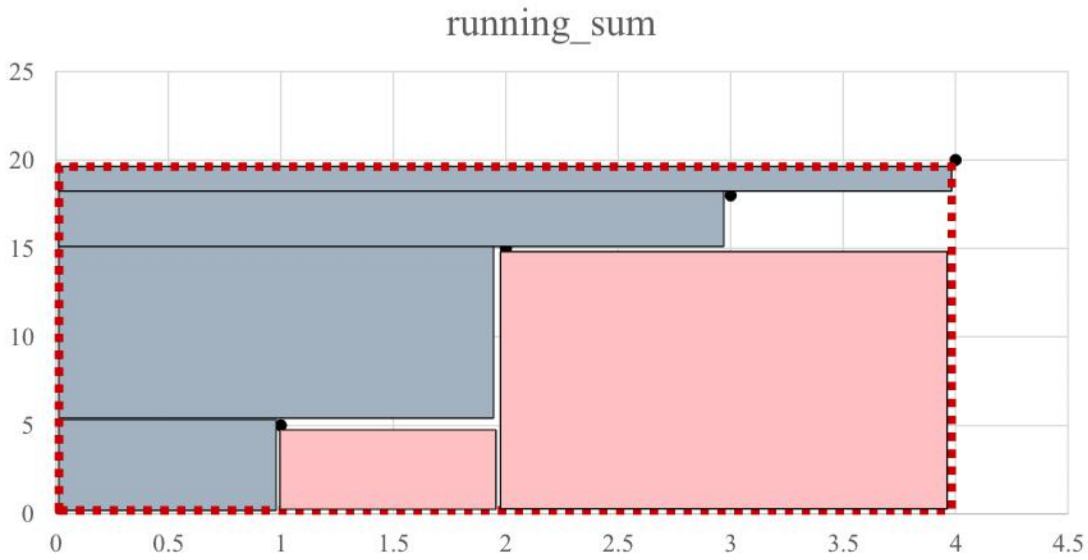
自然语言处理理论基础

- mini-batch在RNN上问题
  - 句子的长度不一样：增加padding



自然语言处理理论基础

- buckets = [2,4,1]；最好的buckets = [1,2,4]

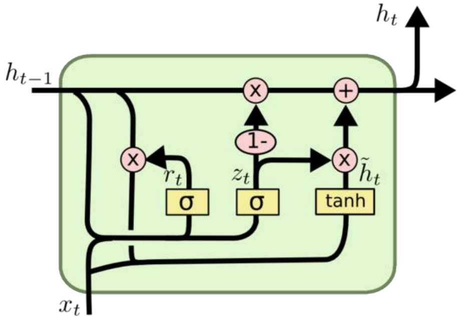


用到的高级API详解

tf.contrib.rnn.GRUCell()

Args:

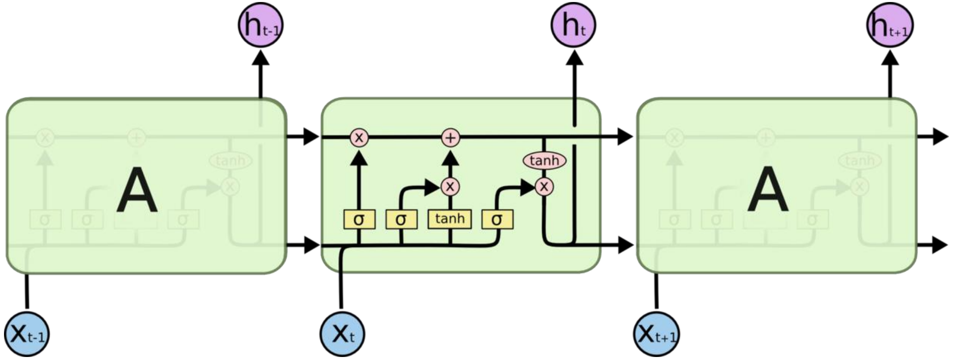
•num_units
•activation
•reuse
•kernel_initializer
•bias_initializer
•name
•dtype



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

tf.contrib.rnn.BasicLSTMCel()

•num_units
•forget_bias
•state_is_tuple
•activation
•reuse
•name
•dtype



知识链接：https://www.cnblogs.com/wangduo/p/6773601.html?utm_source=itdadao&utm_medium=referral

用到的高级API详解

tf.contrib.rnn.MultiRNNCell()

cells: list of RNNCells that will be composed in this order.
state_is_tuple: If True, accepted and returned states are n-tuples, where n = len(cells).
If False, the states are all concatenated along the column axis. This latter behavior will soon be deprecated

用到的高级API详解

tf.contrib.legacy_seq2seq.embedding_attention_seq2seq

Args:

•**encoder_inputs**: A list of 1D int32 Tensors of shape [batch_size].
•**decoder_inputs**: A list of 1D int32 Tensors of shape [batch_size].
•**cell**: tf.nn.rnn_cell.RNNCell defining the cell function and size.
•**num_encoder_symbols**: Integer; number of symbols on the encoder side.
•**num_decoder_symbols**: Integer; number of symbols on the decoder side.
•**embedding_size**: Integer, the length of the embedding vector for each symbol.
•**num_heads**: Number of attention heads that read from attention_states.
•**output_projection**: None or a pair (W, B) of output projection weights and biases; W has shape [output_size x num_decoder_symbols] and B has shape [num_decoder_symbols]; if provided and feed_previous=True, each fed previous output will first be multiplied by W and added B.
•**feed_previous**: Boolean or scalar Boolean Tensor; if True, only the first of decoder_inputs will be used (the "GO" symbol), and all other decoder inputs will be taken from previous outputs (as in embedding_rnn_decoder). If False, decoder_inputs are used as given (the standard decoder case).
•**dtype**: The dtype of the initial RNN state (default: tf.float32).
•**scope**: VariableScope for the created subgraph; defaults to "embedding_attention_seq2seq".
•**initial_state_attention**: If False (default), initial attentions are zero. If True, initialize the attentions from the initial state and attention states.

用到的高级API详解

tf.contrib.legacy_seq2seq.model_with_buckets

Args:

- **encoder_inputs**: A list of Tensors to feed the encoder; first seq2seq input.
- **decoder_inputs**: A list of Tensors to feed the decoder; second seq2seq input.
- **targets**: A list of 1D batch-sized int32 Tensors (desired output sequence).
- **weights**: List of 1D batch-sized float-Tensors to weight the targets.
- **buckets**: A list of pairs of (input size, output size) for each bucket.
- **seq2seq**: A sequence-to-sequence model function; it takes 2 input that agree with encoder_inputs and decoder_inputs, and returns a pair consisting of outputs and states (as, e.g., basic_rnn_seq2seq).
- **softmax_loss_function**: Function (labels, logits) -> loss-batch to be used instead of the standard softmax (the default if this is None). **Note that to avoid confusion, it is required for the function to accept named arguments.**
- **per_example_loss**: Boolean. If set, the returned loss will be a batch-sized tensor of losses for each sequence in the batch. If unset, it will be a scalar with the averaged loss from all examples.
- **name**: Optional name for this operation, defaults to "model_with_buckets".

用到的高级API详解

tf.train.GradientDescentOptimizer(self.learning_rate)

Args:
learning_rate: A Tensor or a floating point value. The learning rate to use.
use_locking: If True use locks for update operations.
name: Optional name prefix for the operations created when applying gradients. Defaults to "GradientDesce

apply_gradients()

grads_and_vars: List of (gradient, variable) pairs as returned by compute_gradients().
global_step: Optional Variable to increment by one after the variables have been updated.
name: Optional name for the returned operation. Default to the name passed to the Optimizer constructor.

minimize()

Args:
- **loss**: A Tensor containing the value to minimize.
- **global_step**: Optional Variable to increment by one after the variables have been updated.
- **var_list**: Optional list or tuple of Variable objects to update to minimize loss. Defaults to the list of variables collected in the graph under the key GraphKeys.TRAINABLE_VARIABLES.
- **gate_gradients**: How to gate the computation of gradients. Can be GATE_NONE, GATE_OP, or GATE_GRAPH.
- **aggregation_method**: Specifies the method used to combine gradient terms. Valid values are defined in the class AggregationMethod.
- **colocate_gradients_with_ops**: If True, try colocating gradients with the corresponding op.
- **name**: Optional name for the returned operation.
- **grad_loss**: Optional. A Tensor holding the gradient computed for loss.

用到的高级API详解

tf.clip_by_global_norm()

Args:

•**t_list**: A tuple or list of mixed Tensors, IndexedSlices, or None.

•**clip_norm**: A 0-D (scalar) Tensor > 0. The clipping ratio.

•**use_norm**: A 0-D (scalar) Tensor of type float (optional). The global norm to use. If not provided, global_norm() is used to compute the norm.

•**name**: A name for the operation (optional)

项目和代码设计思路

Web应用

执行器

数据预处理器 | Seq2Seq_Model

数据基本处理器

Talk is cheap, Show me the code

问答互动

在所报课的课程页面，
1、点击“全部问题”显示本课程所有学员提问的问题。
2、点击“提问”即可向该课程的老师和助教提问问题。

小象学院——互联网新技术在线教育领航者