

03

Para saber mais: Cálculo do Loss

Quando treinamento uma rede neural é comum acompanhar seu loss, o **loss** geralmente é atualizado em cada época, mas com gensim Word2Vec não há uma maneira direta de se fazer isso.

O método para calcular e armazenar o loss é `model.get_latest_training_loss()`. Porém não se calcula por época, e sim por treinamento completo. Entretanto, podemos driblar este problema configurando uma mensagem de *callback*.

Veja como fazer isso com o seguinte código:

```
from gensim.models.callbacks import CallbackAny2Vec
from gensim.models import Word2Vec

# iniciando a chamada callback
class callback(CallbackAny2Vec):
    def __init__(self):
        self.epoch = 0

    def on_epoch_end(self, model):
        loss = model.get_latest_training_loss()
        if self.epoch == 0:
            print('Loss após a época {}: {}'.format(self.epoch, loss))
        else:
            print('Loss após a época {}: {}'.format(self.epoch, loss- self.loss_previous_step))
        self.epoch += 1
        self.loss_previous_step = loss
```

Agora, é preciso configurar alguns Hiperparâmetros na chamada do treinamento:

```
w2v_modelo.train(lista_lista_tokens,
                  total_examples=w2v_modelo.corpus_count,
                  epochs = 30,
                  compute_loss = True,
                  callbacks=[callback()])
```

A cada época, o loss é calculado e exibido na tela. Esse código é inspirado no código que deve entrar em produção no gensim, que vai exibir o loss nas mensagens de Log.

Para mais detalhes e para acompanhar quando a evolução dessa ISSUE, você pode acompanhar a seguinte thread no [github #2135](https://github.com/RaRe-Technologies/gensim/pull/2135) (<https://github.com/RaRe-Technologies/gensim/pull/2135>).

P.S: Caso queira exibir o Loss a cada época, é melhor não exibir as mensagens de Log, evitando uma poluição visual muito grande.