

# Modelo de Linguagem para Português com ULMFit

Monique Monteiro – [moniquebm@tcu.gov.br](mailto:moniquebm@tcu.gov.br)

# Agenda

- Visão geral dos elementos da solução
  - AWD-LSTM
  - ULMFit
- Experimentos
- Super-convergência & 1cycle
- Próximos passos
- Contribuições recentes
- Referências

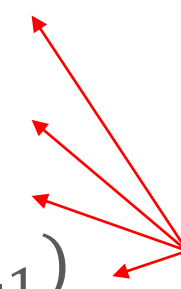
# AWD-LSTM

- Referência:
  - Stephen Merity, Nitish Shirish Keskar and Richard Socher. *Regularizing and Optimizing LSTM Language Models*. 2017.
- Estratégias para regularizar/otimizar LSTM
- DropConnect, *Embedding dropout*
- SGD
- Regularização L2 temporal e nas ativações
- Modelo LSTM 3 camadas
  - 1150 unidades na camada oculta
  - *Embeddings size* = 400

# AWD-LSTM

$$\begin{aligned}i_t &= \sigma(W^i x_t + U^i h_{t-1}) \\f_t &= \sigma(W^f x_t + U^f h_{t-1}) \\o_t &= \sigma(W^o x_t + U^o h_{t-1}) \\\sim c_t &= \tanh(W^c x_t + U^c h_{t-1}) \\c_t &= i * \sim c_t + f_t * \sim c_{t-1} \\h_t &= o_t * \tanh(c_t)\end{aligned}$$

Dropout

A diagram consisting of four red arrows pointing from the word 'Dropout' to the recurrent connection terms  $U^i$ ,  $U^f$ ,  $U^o$ , and  $U^c$  in the equations above. The arrows originate from a single point on the right and point towards each of the four terms.

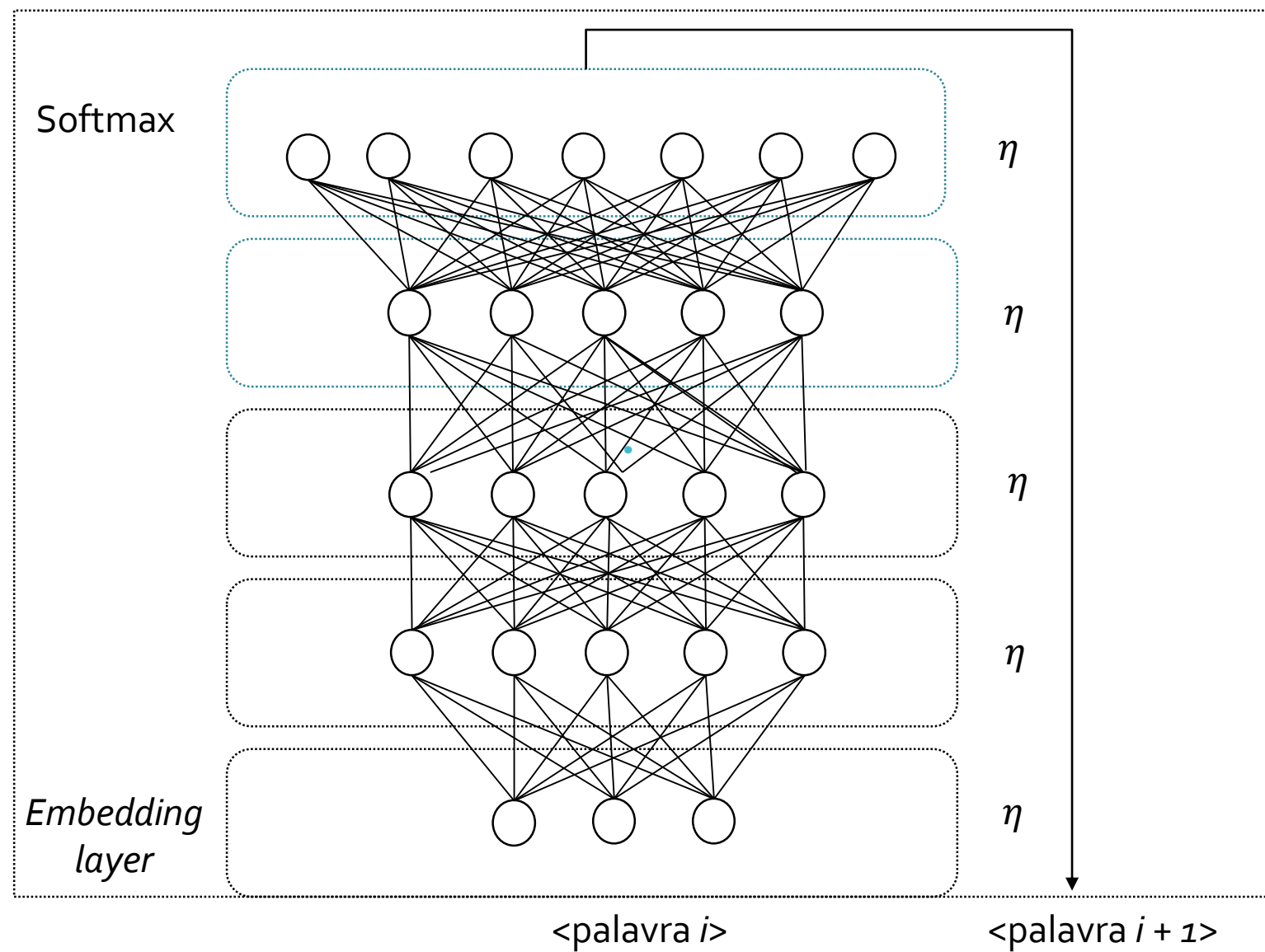
## ULMFit (original)

- Referência:
  - Jeremy Howard and Sebastian Ruder. Fine-tuned Language Models for Text Classification. 2018
- Mesma arquitetura de 3 camadas do AWD-LSTM
- 1ª. Etapa:
  - Semelhante a AWD-LSTM
- 2ª. Etapa:
  - *Discriminative fining-tuning*
  - *Slanted triangular learning rate*

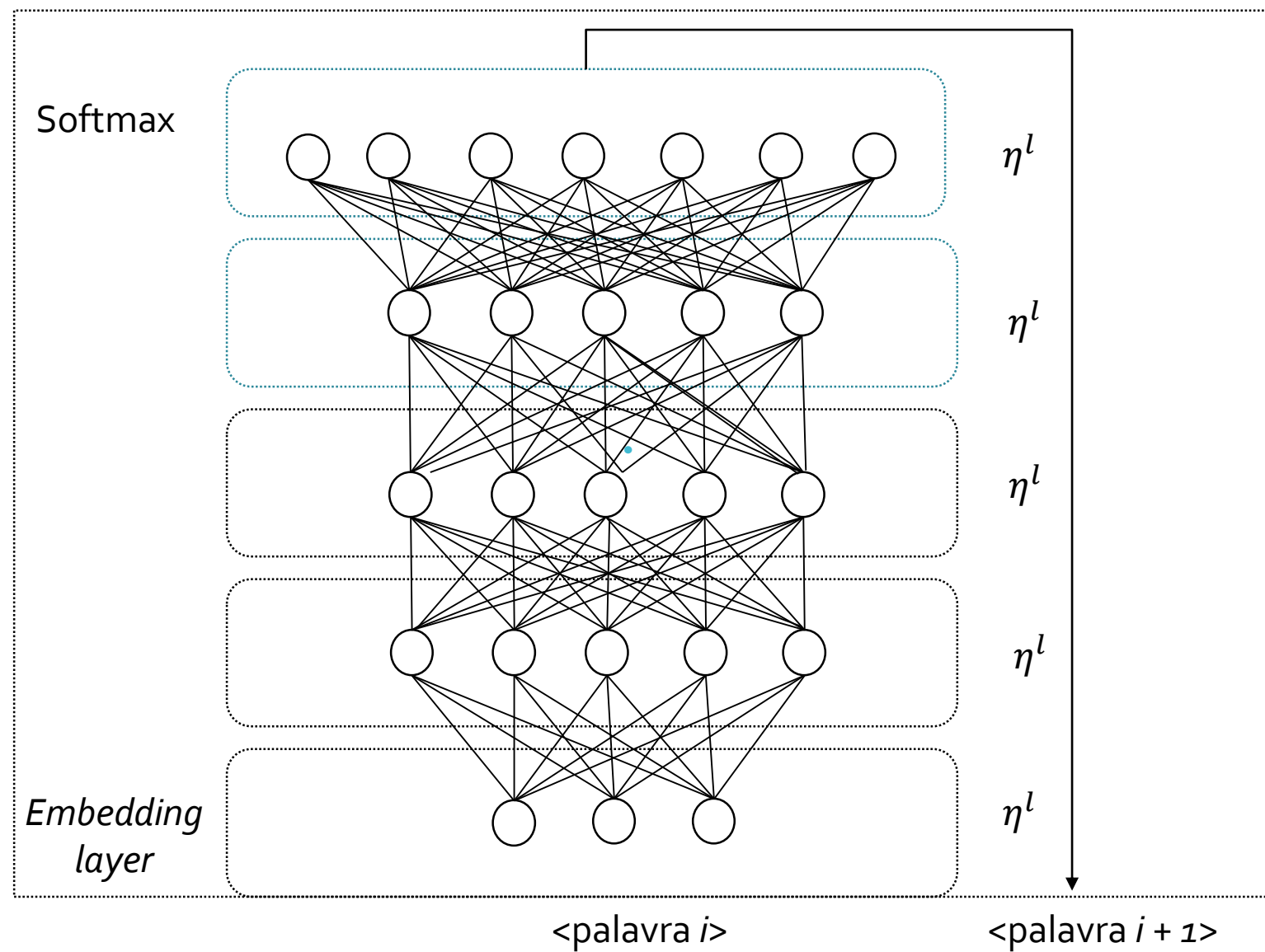
# ULMFit (original)

- 3ª. Etapa:
  - *Discriminative fining-tuning*
  - *Slanted triangular learning rate*
  - *Gradual unfreezing*
  - *Ensemble* bidirecional dos language models
  - 2 novos blocos:
    - *Batch normalization*
    - Dropout
  - ReLU (camadas intermediárias)
  - Softmax (última camada)
  - *Concat pooling*

# ULMFit (original) – 1ª. etapa

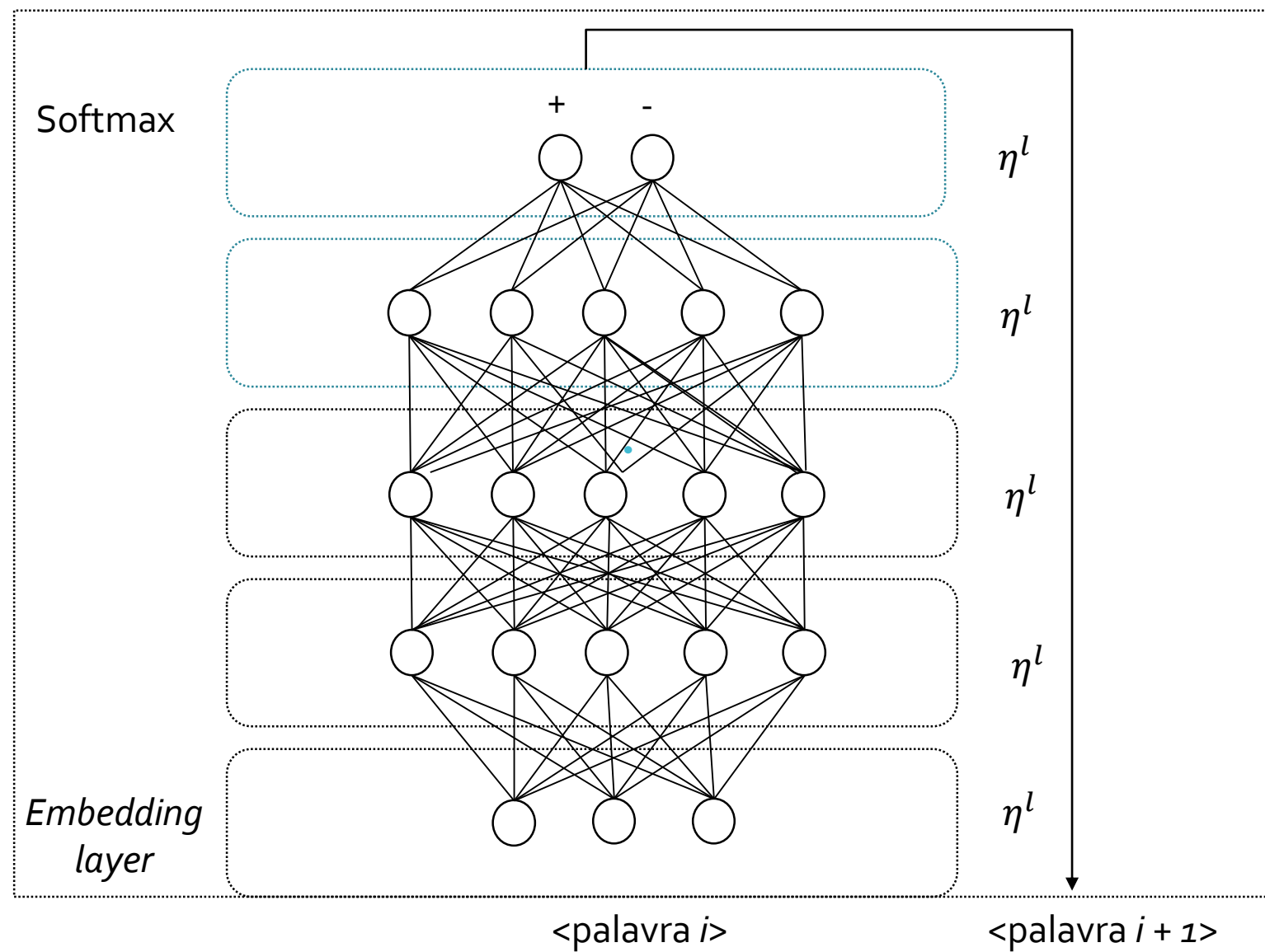


# ULMFit (original) – 2<sup>a</sup>. etapa





# ULMFit (original) – classificador



# Experimentos

- 1º. Treinamento:
  - Número de *epochs*: 1
  - Tamanho do vocabulário: 60000 tokens
  - Frequência mínima de tokens: 5
  - Taxa de aprendizado: 0,0005
  - *Batch size*: 32
  - *Dropouts*: ([0.25, 0.1, 0.2, 0.02, 0.15])\*0.7
  - *Weitgh decay*: 1e-7
  - Bptt: 70
  - Otimizador: Adam, betas=(0.8, 0.99)
  - **Erro de validação: 4.408578**
  - **Perplexidade: 82.15**
  - **Acurácia: 0.26**
  - Resultado do *learning rate finder*: melhor taxa aprox. 0,0005

# Experimentos

- 2º. Treinamento:
  - *Dropouts*:  $([0.25, 0.1, 0.2, 0.02, 0.15]) * 0.05$
  - Erro de validação: **4.73682**
  - Perplexidade: **114.07**
  - Acurácia: **0.26**
  - Resultado do *learning rate finder*: melhor taxa de aprox. **0.001**

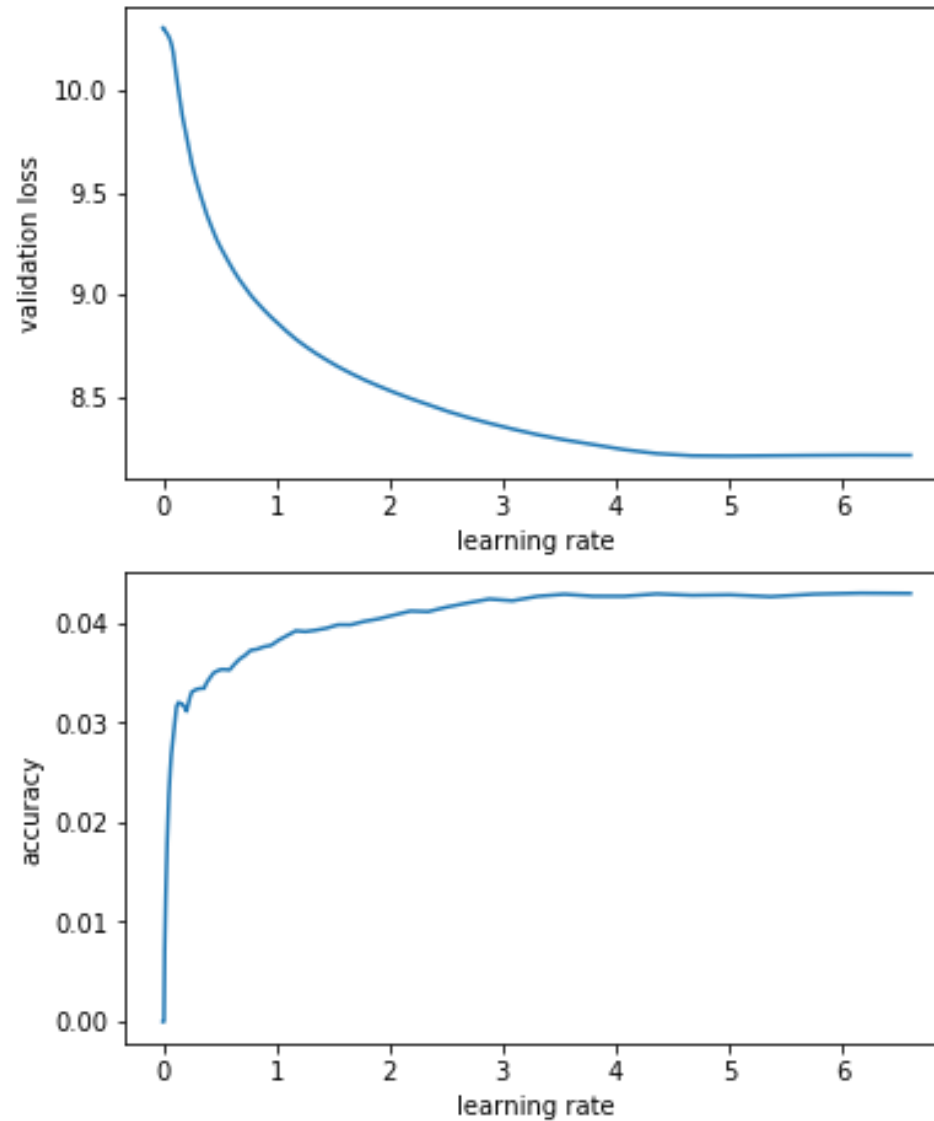
# Experimentos

- 3º. Treinamento:
  - Apenas para encontrar a melhor taxa de aprendizagem
  - Redução do vocabulário para 30000 tokens
  - *Batch size*: 52
  - Resultado do *learning rate finder*: melhor taxa de aprox. 0.001
  - Erro de validação: 3.933865
  - Perplexidade: 51.10
  - Acurácia: 0.28

# Super- convergência

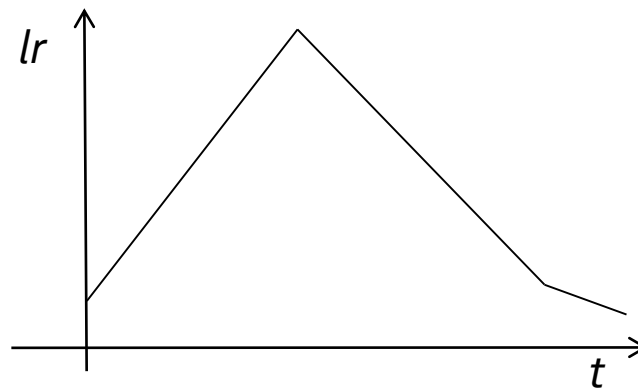
- Taxa de aprendizado muito pequena → *overfitting*
- Taxa de aprendizado alta → regularização
- *Cyclical learning rate*:
  - Valor mínimo e valor máximo
  - Ciclo: crescimento + decrescimento, linear
  - *LR range test* → define o valor máximo
- *Super-convergence*:
  - Altas taxas de aprendizado
  - Regularização
  - Treinamento mais rápido

# Super- convergência



# 1cycle

- Referência:
  - Leslie N. Smith. A disciplined approach to neural network hyper-parameters: Part 1 - learning rate, batch size, momentum, and weight decay. 2018
- *1cycle*:
  - Ciclo < número total de iterações/epochs
  - Iterações restantes: decrescimento
  - Acurácia estabiliza antes do fim do treinamento
  - Combinação de outras técnicas conhecidas



## Experimentos (cont.)

- 4º. Treinamento: *1cycle*
  - Apenas para encontrar a melhor taxa de aprendizagem
  - Número de *epochs*: 2
  - Otimizador: SGD com momentum = 0.9
  - **Erro de validação: 3.618101**
  - **Perplexidade: 37.27**
  - **Acurácia: 0.30**
  - **Melhor taxa de aprendizado encontrada: aprox. 5.0**



## Experimentos (cont.)

- Treinamento final:
  - Número de *epochs*: 10
  - *Batch size*: 52
  - *Weight-decay*:  $1e-7$
  - Bptt: 70
  - *Dropouts*:  $([0.25, 0.1, 0.2, 0.02, 0.15]) * 0.05$
  - SGD com momentum=0.9
  - Taxa de aprendizado: 5.0
  - **Erro de validação: 3.465898**
  - **Perplexidade: 32**
  - **Acurácia: 0.32**

## Experimentos (cont.)

- Experimento adicional:
  - Aumento da taxa de *dropout* para 0.1
  - Uso de *gradient clipping* = 0.25
  - Erro de validação: 3.611512

## Dicas do fórum do fast.ai ("Language Model Zoo")

- Limitar o corpus a 100 milhões de tokens
- Redução do tamanho do vocabulário de 60000 para 30000 tokens
  - Tempo médio de treinamento para cada epoch caiu de 3 para 2 horas.
  - Menor consumo de memória
- 1cycle / *super-convergence*
  - use\_clr\_beta (fastai)
    - *Momentum* cíclico

## Próximos passos

- Treinamento do *language model* específico:
  - Documentos públicos de controle externo do TCU
- Estudo de caso com classificador de assuntos
- Otimizações no *language model*:
  - *Continuous cache pointer*
  - *Quasi-recurrent neural networks* (QRNN)

Para manter  
no radar...

- Trieu H. Trinh, Quoc V. Le . [A Simple Method for Commonsense Reasoning](#). (*Submitted on 7 Jun 2018*)
- Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever. [Improving Language Understanding with Unsupervised Learning](#)
- [The Natural Language Decathlon](#)

## Outras referências

- [Language Model for Telugu \(Indian\) Language](#)
- ["1cycle" Sgugger's posts](#) (incluindo hiperparâmetros)
- [Scripts no Github](#)

# Modelo de Linguagem para Português com ULMFit

Monique Monteiro – [moniquebm@tcu.gov.br](mailto:moniquebm@tcu.gov.br)