Memória em Curvas Elípticas

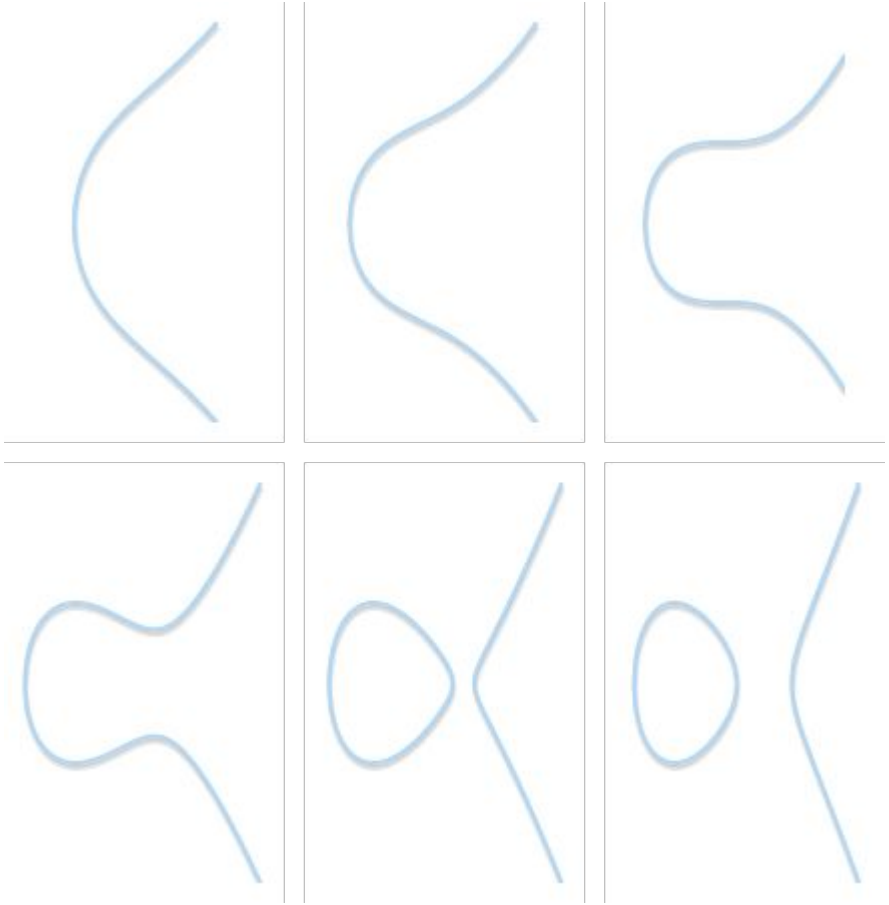
Manoel Domingues Junior

COPPE UFRJ
Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa de Engenharia

Curvas Elípticas

- Não são elipses
- São utilizadas em criptografia assimétrica

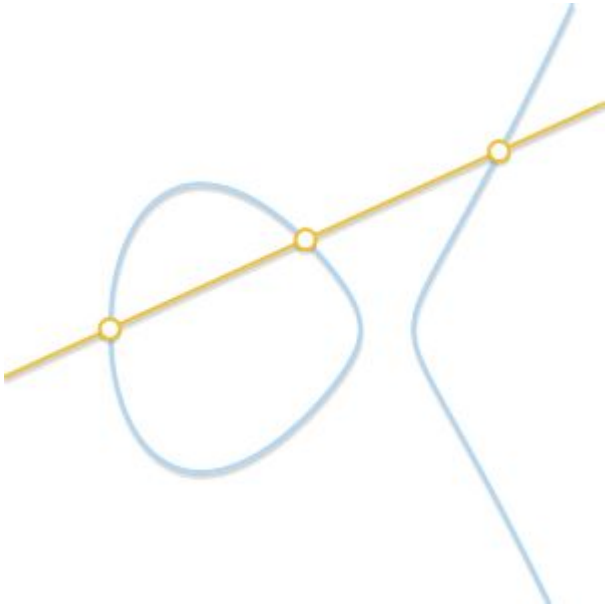
$$y^2 = x^3 + ax + b$$



https://andrea.corbellini.name/2015/05/17/elliptic-curve-cryptography-a-gentle-introduction/

COPPE UFRJ
Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa de Engenharia

Curvas Elípticas

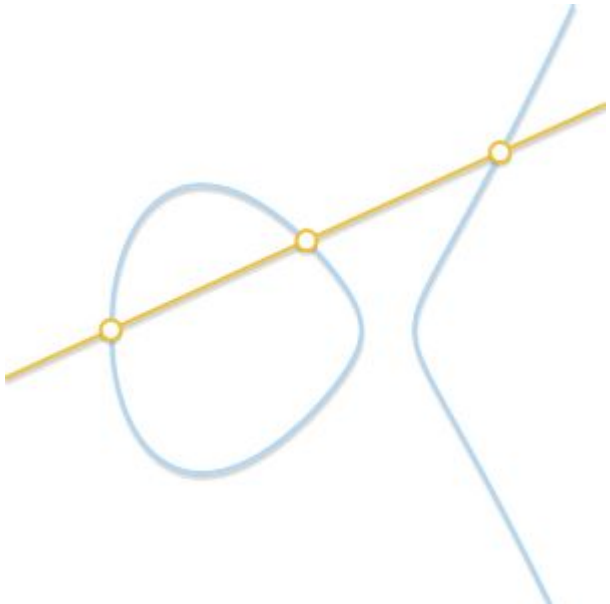


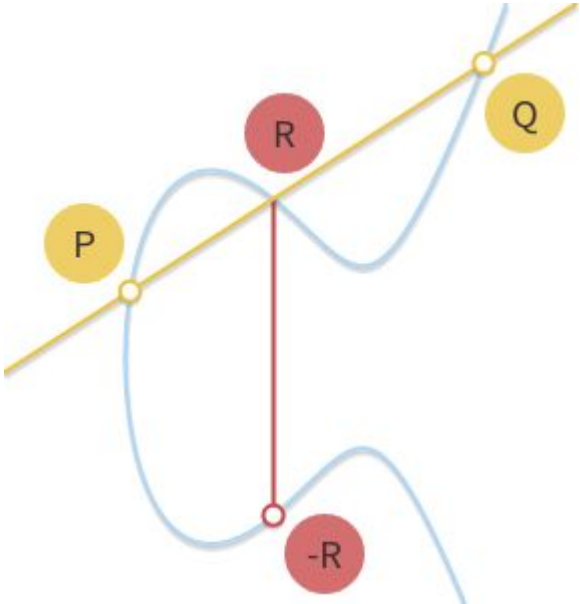
https://andrea.corbellini.name/2015/05/17/elliptic-curve-cryptography-a-gentle-introduction/

Curvas Elípticas

$P + R + Q = 0$

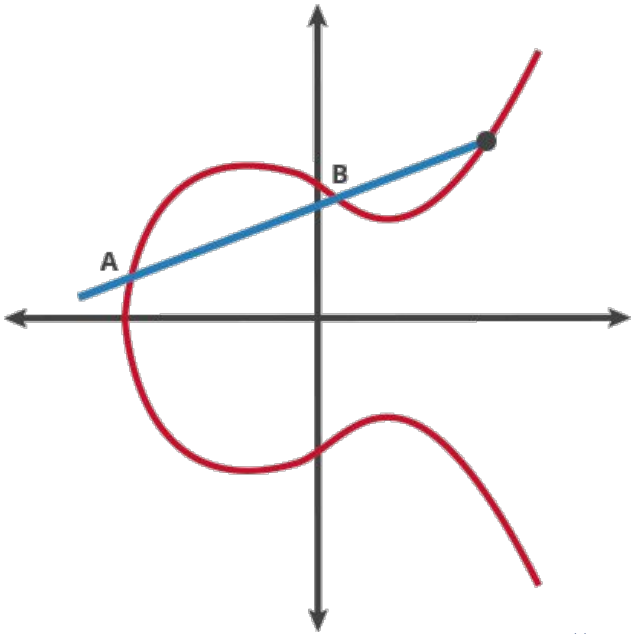
$P + Q = - R$

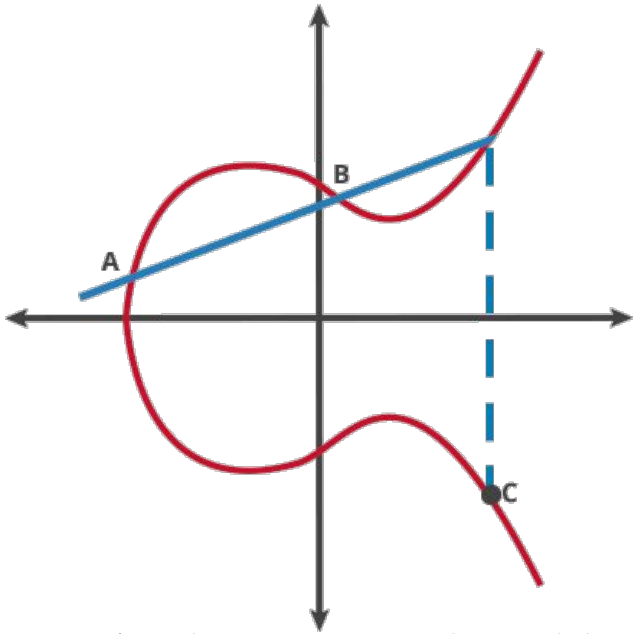




https://andrea.corbellini.name/2015/05/17/elliptic-curve-cryptography-a-gentle-introduction/

Curvas Elípticas





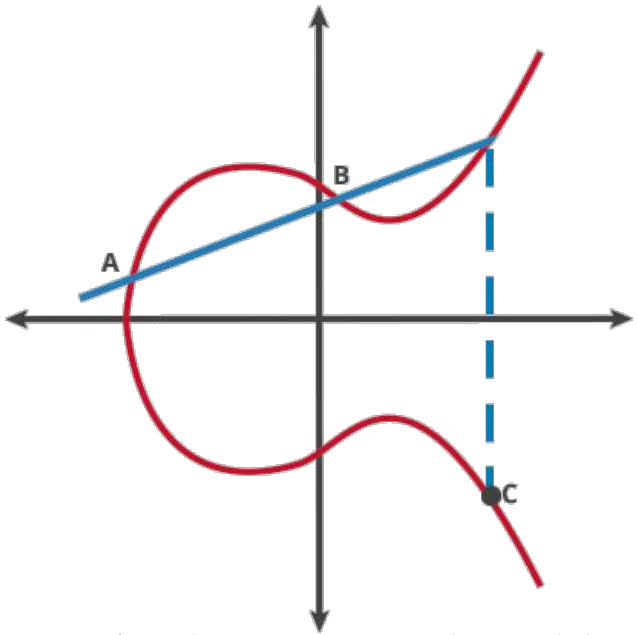
A + B = C

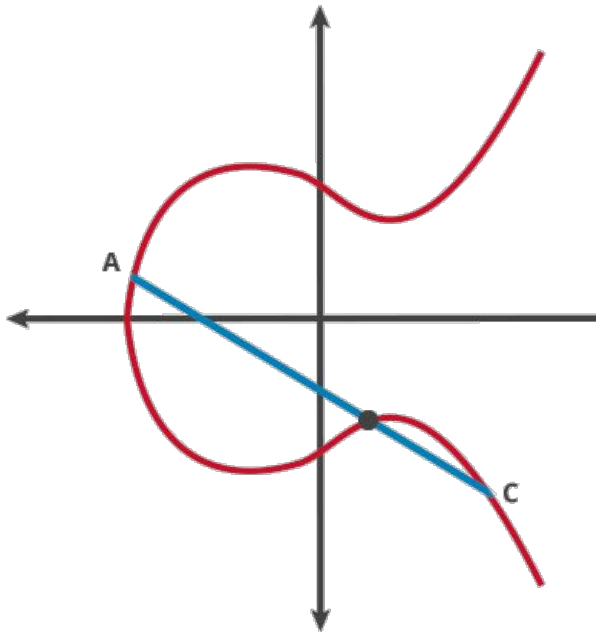
https://blog.cloudflare.com/a-relatively-easy-to-understand-primer-on-elliptic-curve-cryptography/

Curvas Elípticas

A + B = C

A + C = D



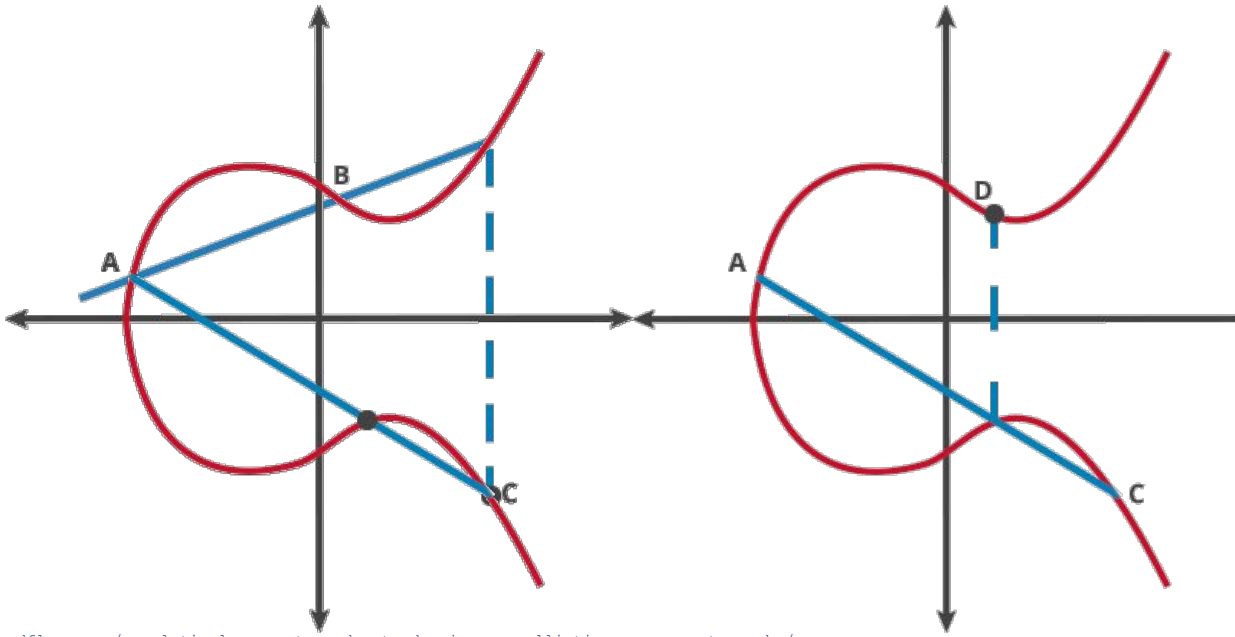


https://blog.cloudflare.com/a-relatively-easy-to-understand-primer-on-elliptic-curve-cryptography/

Curvas Elípticas

A + B = C

A + C = D



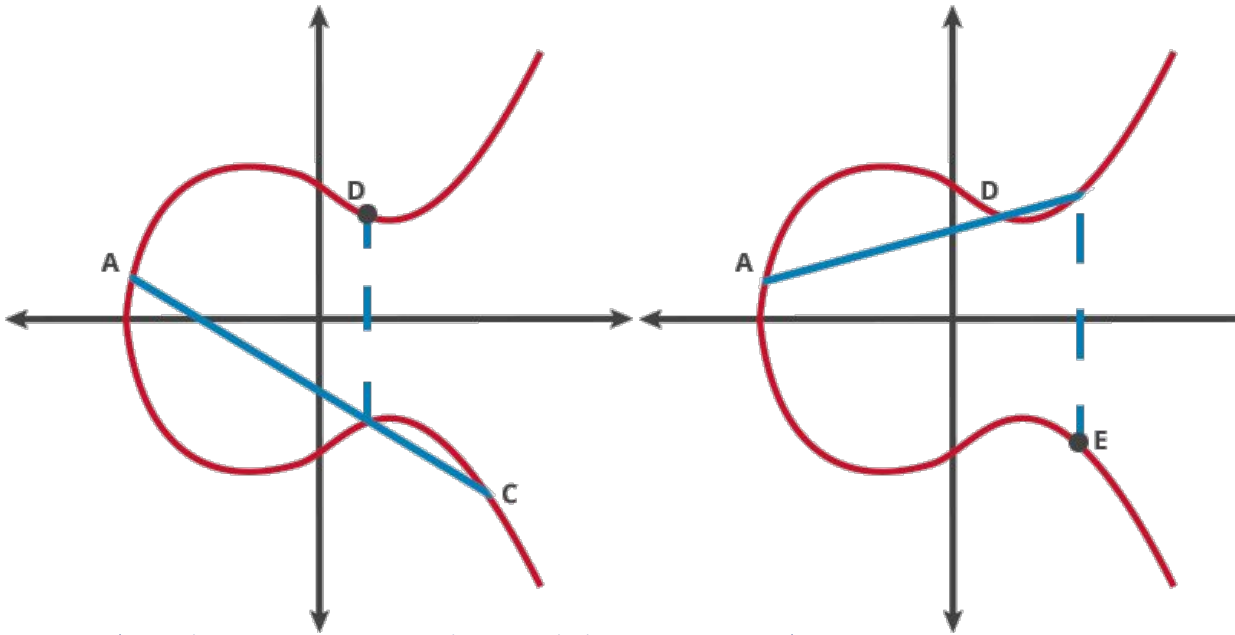
https://blog.cloudflare.com/a-relatively-easy-to-understand-primer-on-elliptic-curve-cryptography/

Curvas Elípticas

A + B = C

A + C = D

A + D = E



https://blog.cloudflare.com/a-relatively-easy-to-understand-primer-on-elliptic-curve-cryptography/

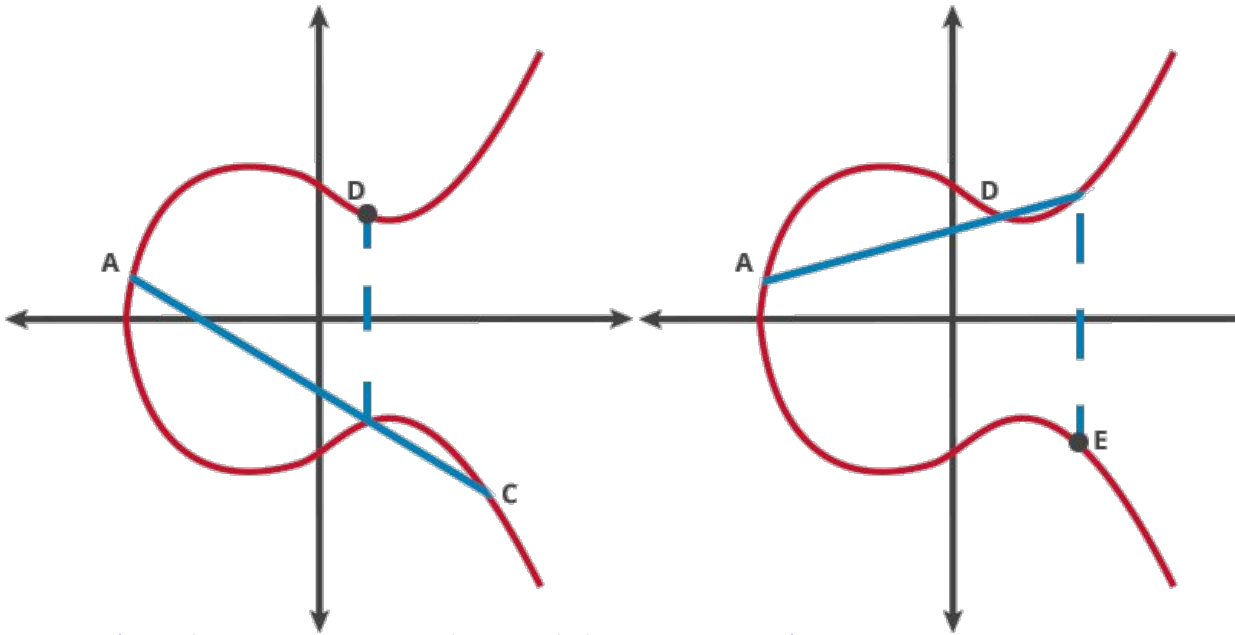
Curvas Elípticas

A + A = B

A + B = C

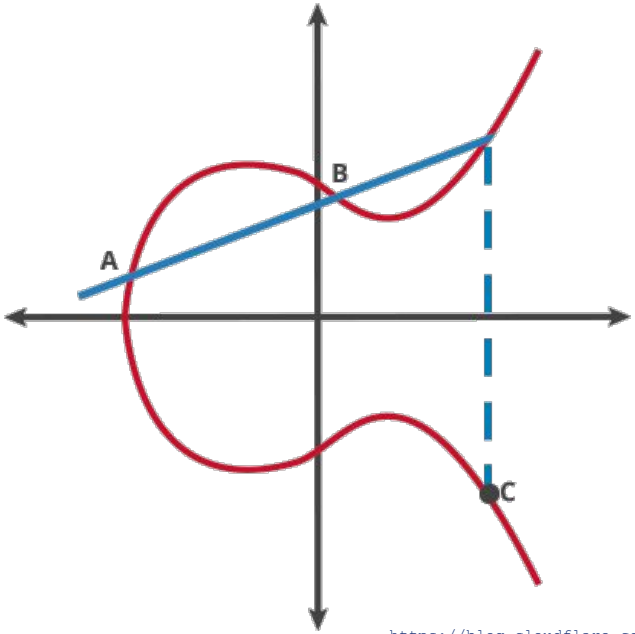
A + C = D

A + D = E



https://blog.cloudflare.com/a-relatively-easy-to-understand-primer-on-elliptic-curve-cryptography/

Curvas Elípticas - caso do trabalho



A - escolhido aleatoriamente

B - fixo (x = 9)

Será que uma rede neural consegue aprender quem seria o **C**?

Vamos dividir o trabalho em partes:
- adição
- mod

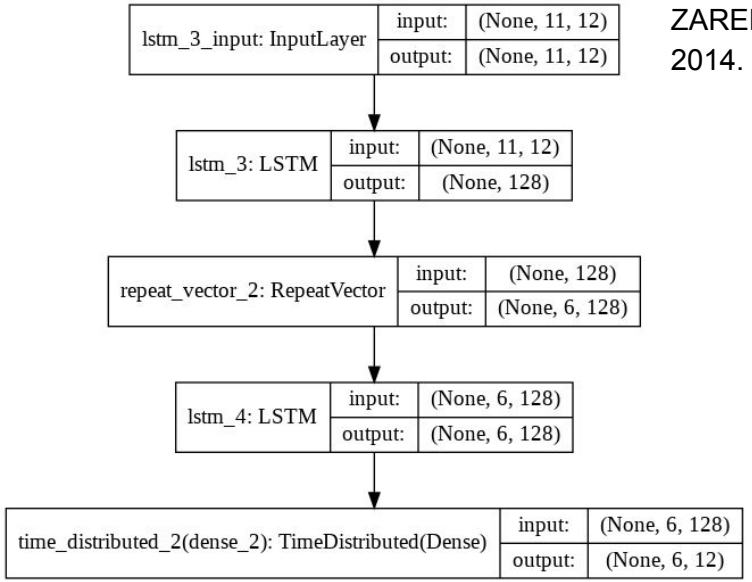
https://blog.cloudflare.com/a-relatively-easy-to-understand-primer-on-elliptic-curve-cryptography/

Rede Neural para adição

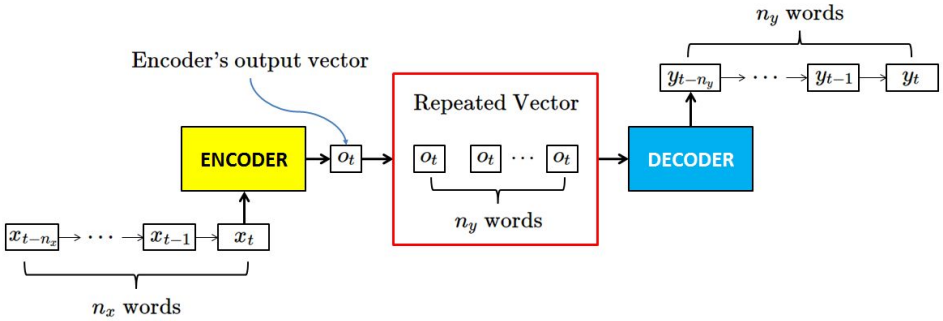
ZAREMBA, Wojciech; SUTSKEVER, Ilya. Learning to Execute. **arXiv [cs.NE]**, 2014. Disponível em: <https://arxiv.org/abs/1410.4615>.

- 2 digits reversed: + 1 layer LSTM (128 HN), 5k training examples = 99% train/test accuracy in 55 epochs
- 3 digits reversed: + 1 layer LSTM (128 HN), 50k training examples = 99% train/test accuracy in 100 epochs
- 4 digits reversed: + 1 layer LSTM (128 HN), 400k training examples = 99% train/test accuracy in 20 epochs
- **5 digits reversed: + 1 layer LSTM (128 HN), 550k training examples = 99% train/test accuracy in 30 epochs**

Rede Neural para adição

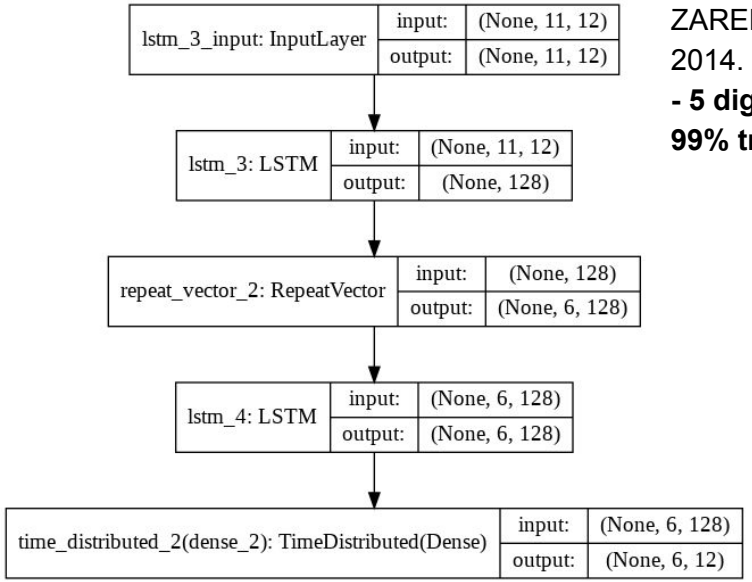


ZAREMBA, Wojciech; SUTSKEVER, Ilya. Learning to Execute. **arXiv [cs.NE]**, 2014. Disponível em: <https://arxiv.org/abs/1410.4615>.



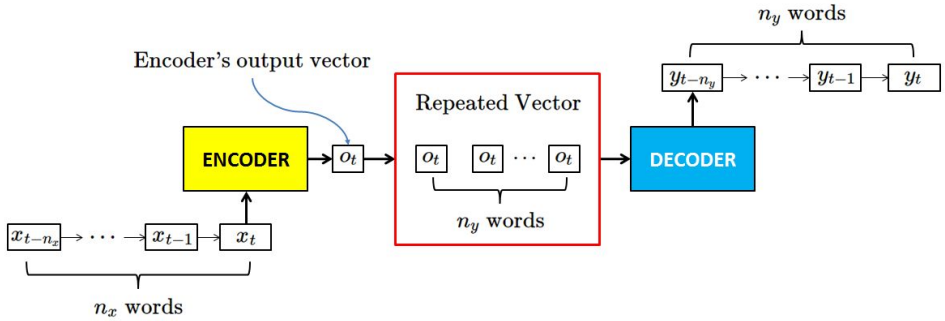
https://chunml.github.io/ChunML.github.io/project/Sequence-To-Sequence/

Rede Neural para adição



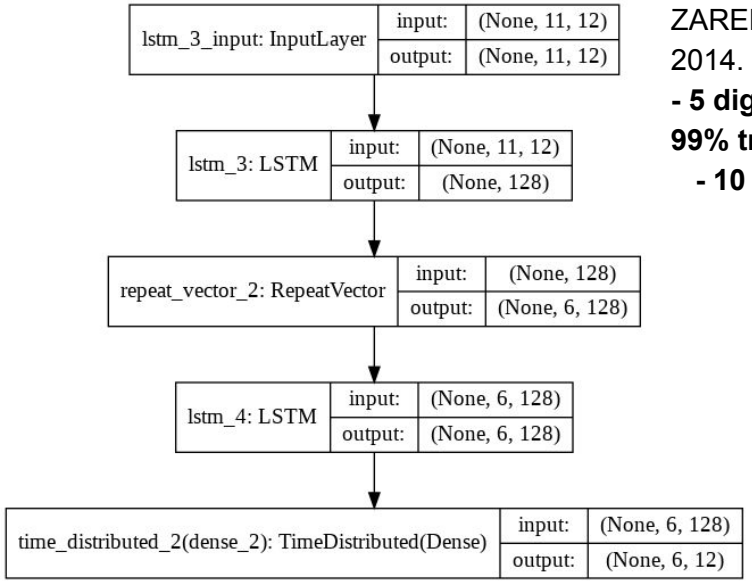
ZAREMBA, Wojciech; SUTSKEVER, Ilya. Learning to Execute. **arXiv [cs.NE]**, 2014. Disponível em: <https://arxiv.org/abs/1410.4615>.

- 5 digits reversed: + 1 layer LSTM (128 HN), 550k training examples = 99% train/test accuracy in 30 epochs



https://chunml.github.io/ChunML.github.io/project/Sequence-To-Sequence/

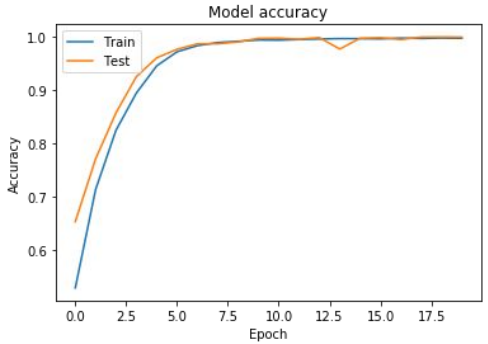
Rede Neural para adição

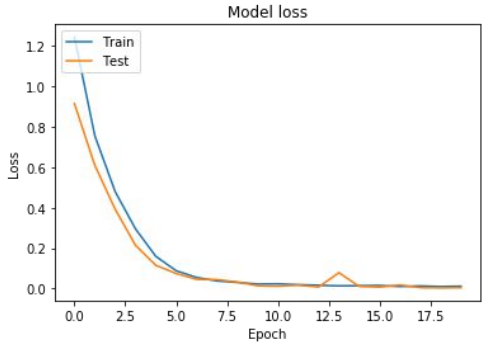


ZAREMBA, Wojciech; SUTSKEVER, Ilya. Learning to Execute. **arXiv [cs.NE]**, 2014. Disponível em: <https://arxiv.org/abs/1410.4615>.

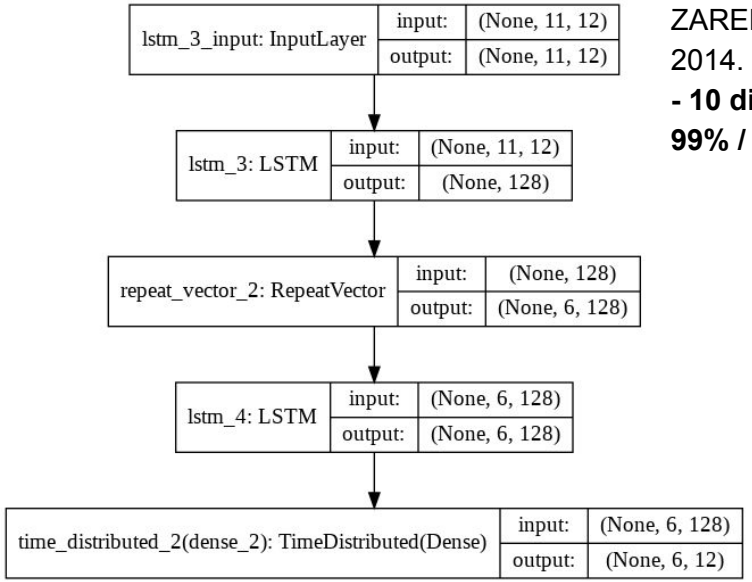
- 5 digits reversed: + 1 layer LSTM (128 HN), 550k training examples = 99% train/test accuracy in 30 epochs

- 10 epochs (suficiente)



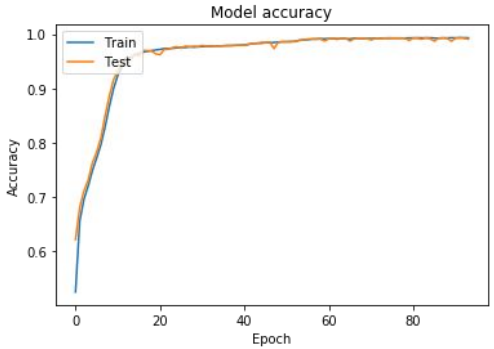


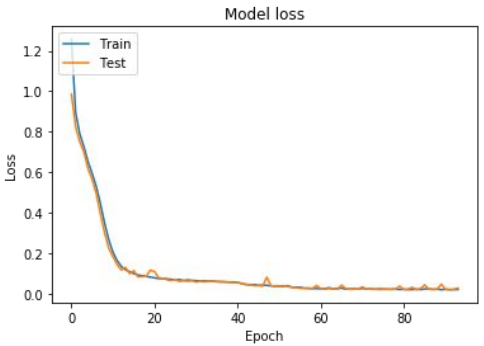
Rede Neural para adição



ZAREMBA, Wojciech; SUTSKEVER, Ilya. Learning to Execute. **arXiv [cs.NE]**, 2014. Disponível em: <https://arxiv.org/abs/1410.4615>.

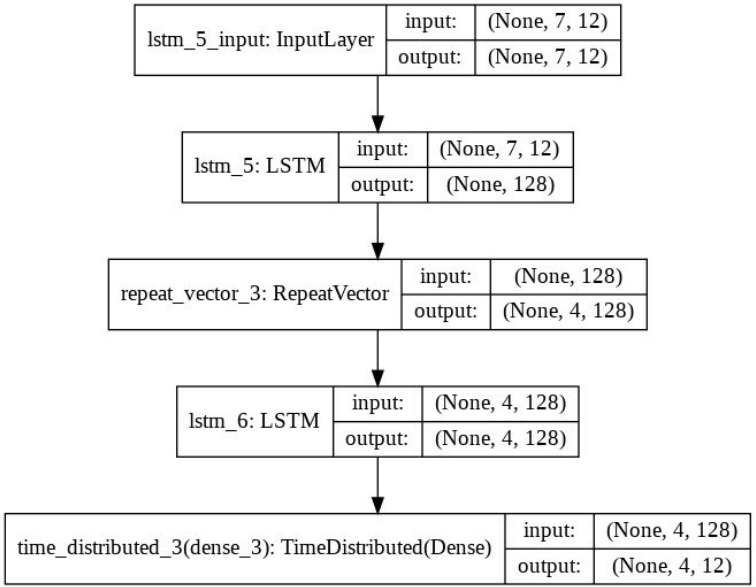
- 10 digits reversed: + 1 layer LSTM (128 HN), 600k training examples = 99% / 99% train/test accuracy in 94 epochs





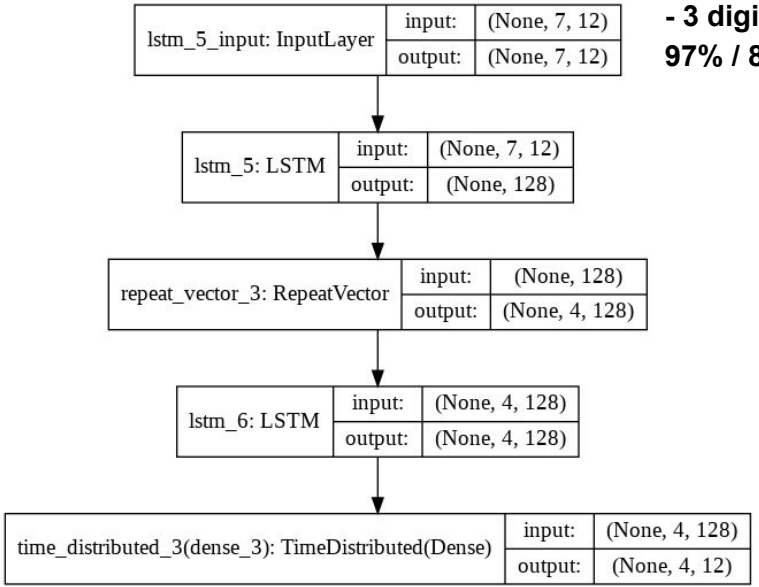
COPPE UFRJ
Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa de Engenharia

Rede Neural para *mod*

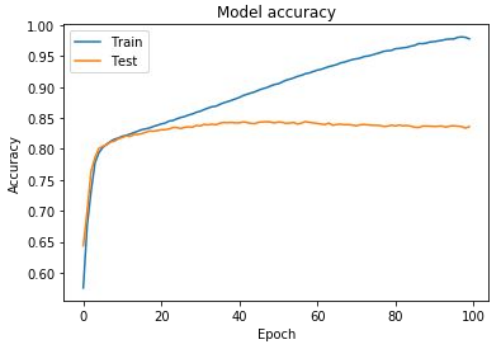


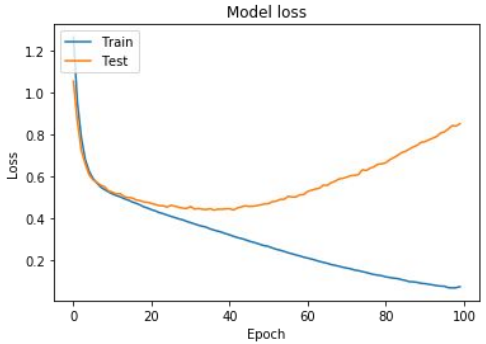
COPPE UFRJ
Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa de Engenharia

Rede Neural para *mod*



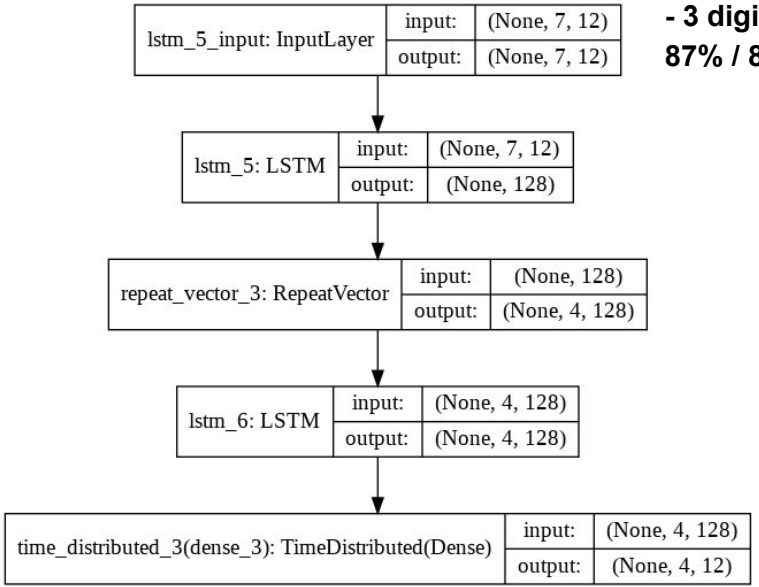
- 3 digits: + 1 layer LSTM (128 HN), 50k training examples = 97% / 83% train/test accuracy in 100 (30) epochs



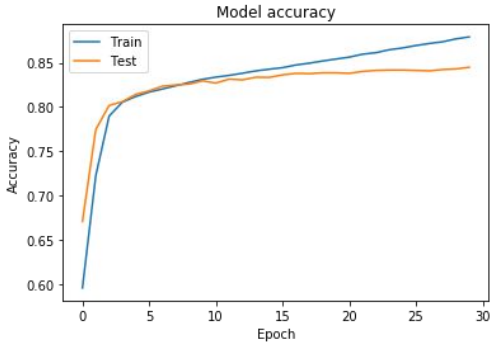


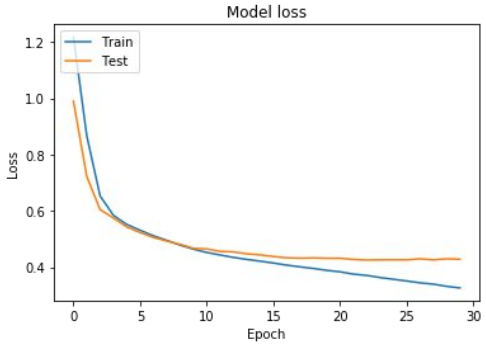
COPPE UFRJ
Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa de Engenharia

Rede Neural para *mod*



- 3 digits reversed: + 1 layer LSTM (128 HN), 50k training examples = 87% / 84% train/test accuracy in 30 epochs





COPPE UFRJ
Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa de Engenharia

Rede Neural para *mod*

```
Q 94%711   T 94    ☑ 94
Q 174%594  T 174   ☑ 174
Q 33%656   T 33    ☑ 33
Q 255%6    T 3     ☑ 3
Q 565%848  T 565   ☑ 565
Q 7%747    T 7     ☑ 7
Q 39%337   T 39    ☑ 39
Q 85%755   T 85    ☑ 85
Q 9%938    T 9     ☑ 9
Q 403%888  T 403   ☑ 403
Q 340%762  T 340   ☑ 340
Q 59%98    T 59    ☑ 59
Q 81%94    T 81    ☑ 81
Q 96%272   T 96    ☑ 96
Q 120%849  T 120   ☑ 120
Q 71%515   T 71    ☑ 71
Q 9%231    T 9     ☑ 9
Q 9%138    T 9     ☑ 9
Q 474%785  T 474   ☑ 474
Q 991%1    T 0     ☑ 0
Q 872%0    T 0     ☑ 0
Q 4%278    T 4     ☑ 4
Q 611%963  T 611   ☑ 611
Q 381%868  T 381   ☑ 381
Q 5%681    T 5     ☑ 5
Q 14%14    T 0     ☑ 0
Q 167%204  T 167   ☑ 167
Q 11%177   T 11    ☑ 11
Q 204%683  T 204   ☑ 204
Q 269%694  T 269   ☑ 269
Q 49%861   T 49    ☑ 49
```

lstm_5_in … ne, 7, 12)
… ne, 7, 12)

lstm_ … , 12)
… 128)

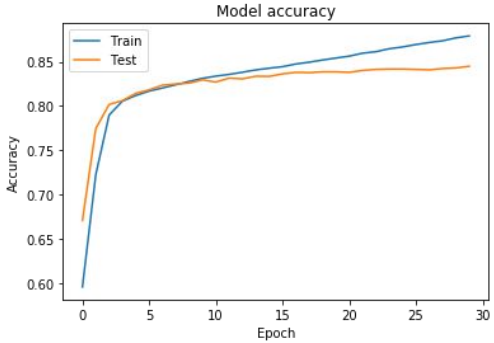
repeat_vecto … None, 128)
… None, 4, 128)

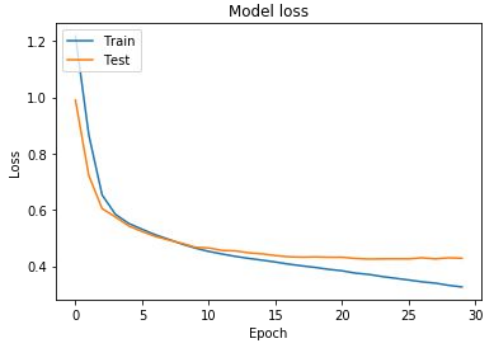
lstm_ … 128)
… 128)

time_distributed_3(dense … input: (None, 4, 128)
… output: (None, 4, 12)

- 3 digits reversed: + 1 layer LSTM (128 HN), 50k training examples = 87% / 84% train/test accuracy in 30 epochs

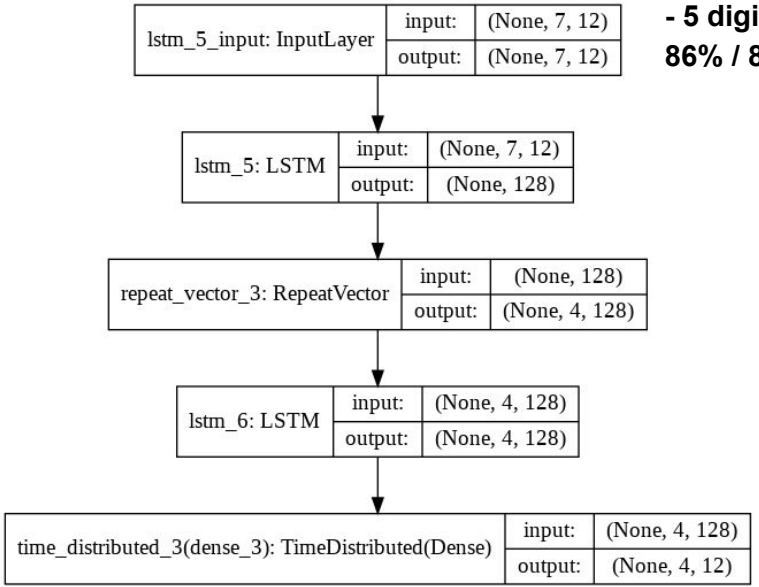
Sem alterações, mas podemos ver que a rede aprende a **repetir o menor número**.



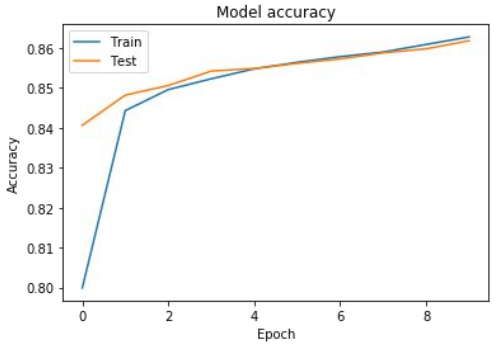


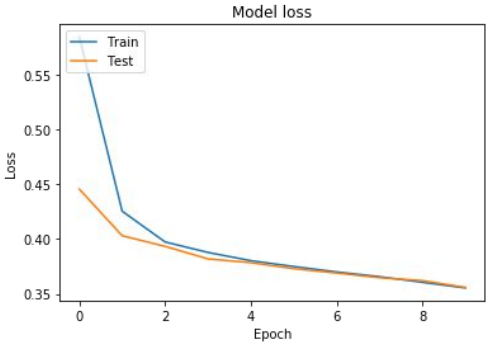
COPPE UFRJ
Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa de Engenharia

Rede Neural para *mod*



- 5 digits reversed: + 1 layer LSTM (128 HN), 500k training examples = 86% / 86% train/test accuracy in 10 epochs





COPPE UFRJ
Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa de Engenharia

Rede Neural para *mod*

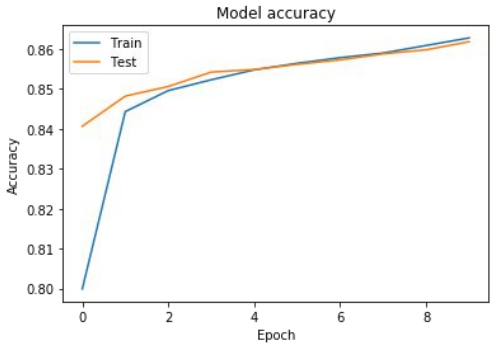
```
Q 56%9167      T 56      ☑ 56
Q 62%9628      T 62      ☑ 62
Q 35313%79136  T 35313   ☑ 35313
Q 3805%2550    T 1255    ☑ 1255
Q 242%42487    T 242     ☑ 242
Q 27706%1      T 0       ☑ 0
Q 3951%68549   T 3951    ☑ 3951
Q 85210%0      T 0       ☑ 0
Q 20632%38371  T 20632   ☑ 20632
Q 1%8514       T 1       ☑ 1
Q 7204%0       T 0       ☑ 0
Q 290%79791    T 290     ☑ 290
Q 4035%9664    T 4035    ☑ 4035
Q 27%7376      T 27      ☑ 27
Q 5028%50989   T 5028    ☑ 5028
Q 548%86176    T 548     ☑ 548
Q 418%77479    T 418     ☑ 418
Q 37%61320     T 37      ☑ 37
Q 9610%61473   T 9610    ☑ 9610
Q 35%69307     T 35      ☑ 35
Q 95725%8      T 5       ☑ 5
Q 27%67161     T 27      ☑ 27
Q 9%432        T 9       ☑ 9
Q 9149%8       T 5       ☑ 5
Q 3042%0       T 0       ☑ 0
Q 60%116       T 60      ☑ 60
Q 1335%9823    T 1335    ☑ 1335
Q 1%3420       T 1       ☑ 1
Q 52%58222     T 52      ☑ 52
```

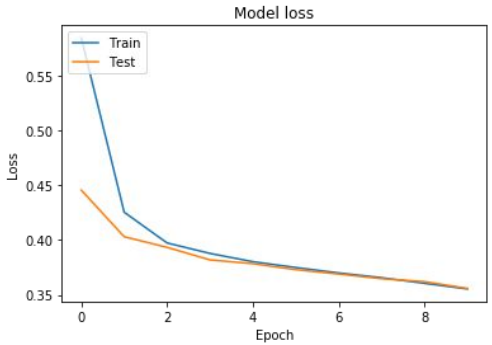
lstm_ 12)
12)

repeat_ , 128)
4, 128)

time_distributed_3(: (None, 4, 128)
t: (None, 4, 12)

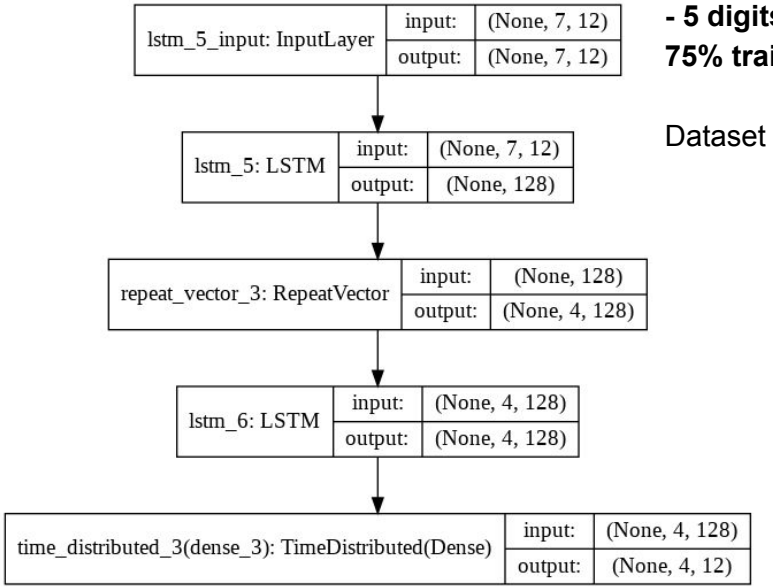
- 5 digits reversed: + 1 layer LSTM (128 HN), 500k training examples = 86% / 86% train/test accuracy in 10 epochs





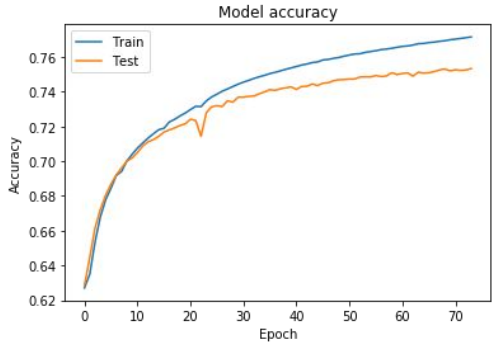
COPPE UFRJ
Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa de Engenharia

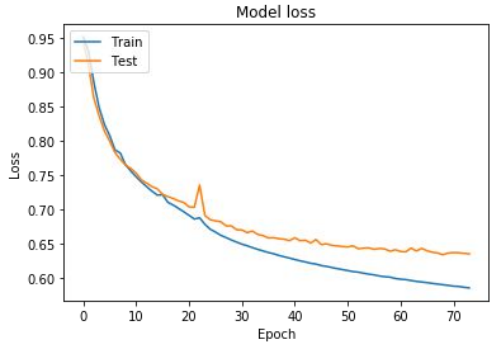
Rede Neural para *mod*



- 5 digits reversed: + 1 layer LSTM (128 HN), 500k training examples = 75% train/test accuracy in 70 epochs

Dataset ajustado para casos onde o resultado é diferente dos operandos.





COPPE UFRJ
Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa de Engenharia

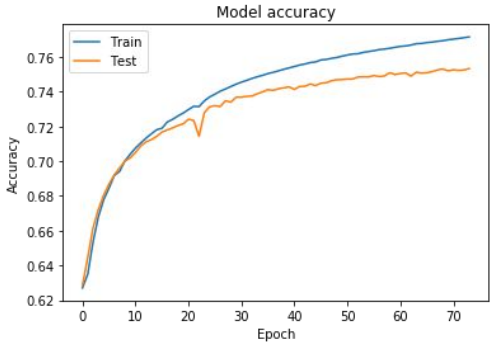
Rede Neural para *mod*

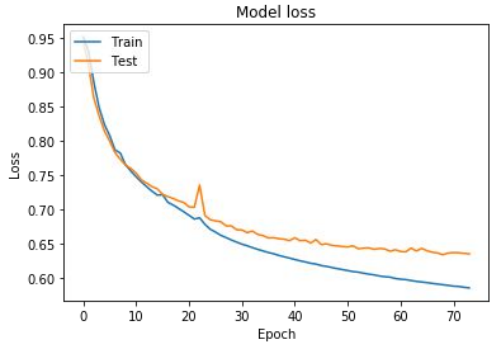
```
Q 8715%7591   T 1124   ☑ 1124
Q 3368%2747   T 621    ☑ 621
Q 9827%3646   T 2535   ☑ 2535
Q 768%608     T 160    ☑ 160
Q 4807%4766   T 41     ☑ 41
Q 250%32      T 26     ☑ 26
Q 912%539     T 373    ☑ 373
Q 1380%860    T 520    ☑ 520
Q 9673%5581   T 4092   ☑ 4092
Q 4756%2505   T 2251   ☑ 2251
Q 787%532     T 255    ☑ 255
Q 312%229     T 83     ☑ 83
Q 768%291     T 186    ☑ 186
Q 8622%6621   T 2001   ☑ 2001
Q 740%697     T 43     ☑ 43
Q 739%98      T 53     ☑ 53
Q 6730%2348   T 2034   ☑ 2034
Q 21215%50    T 15     ☑ 15
Q 679%237     T 205    ☑ 205
Q 352%253     T 99     ☑ 99
Q 9044%5557   T 3487   ☑ 3487
Q 268%69      T 61     ☑ 61
Q 7166%146    T 12     ☑ 12
Q 68291%25    T 16     ☑ 16
Q 637%66      T 43     ☑ 43
```

ls … repe… … time_distributed … (None, 4, 128) (None, 4, 12)

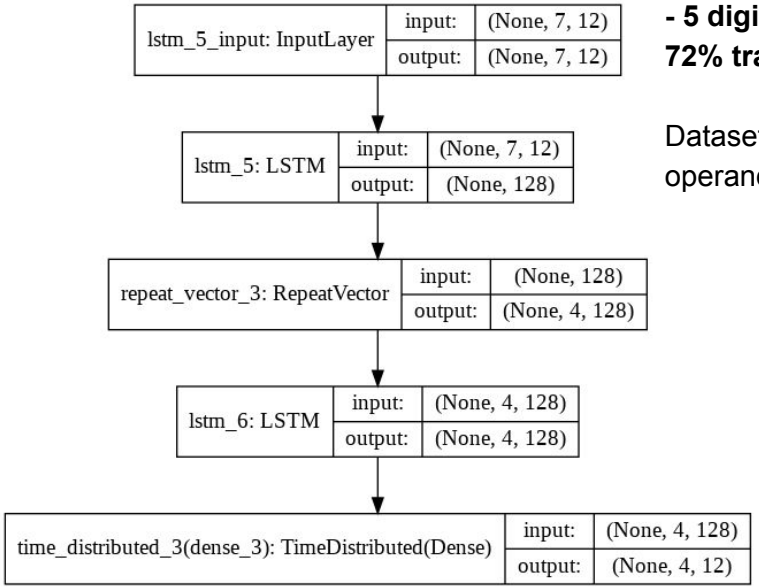
- 5 digits reversed: + 1 layer LSTM (128 HN), 500k training examples = 75% train/test accuracy in 70 epochs

Dataset ajustado para casos onde o resultado é diferente dos operandos.



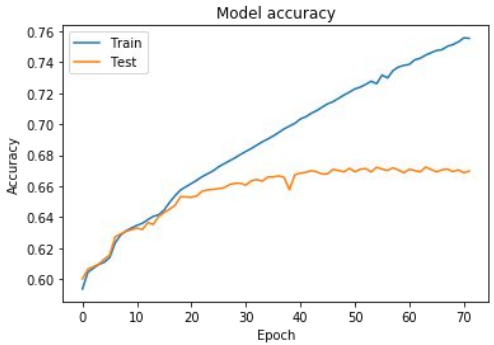


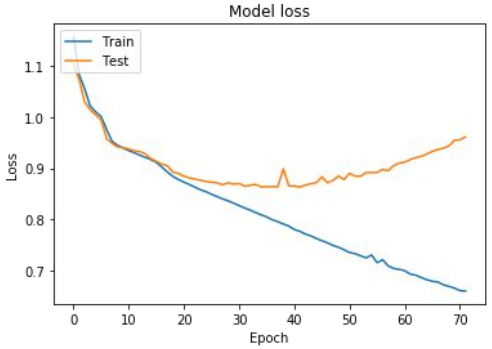
Rede Neural para *mod*



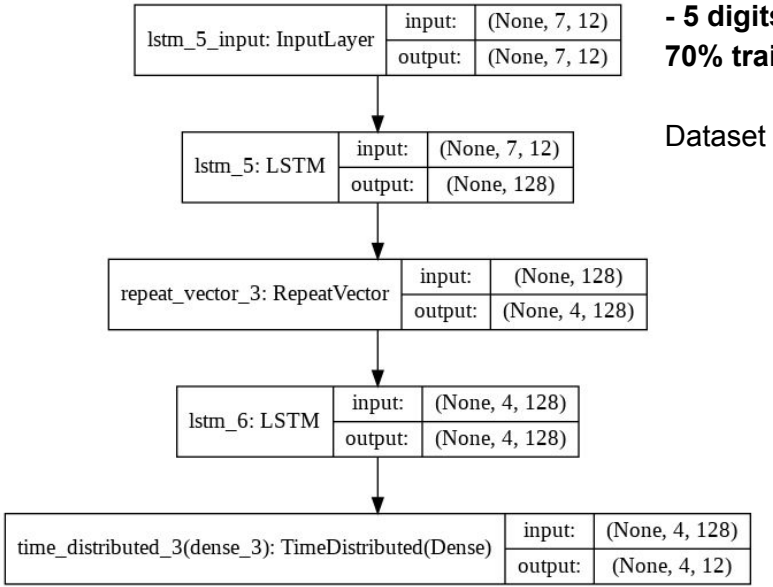
- 5 digits reversed: + 1 layer LSTM (128 HN), 100k training examples = 72% train/test accuracy in 76 epochs

Dataset (menor) ajustado para casos onde o resultado é diferente dos operandos.



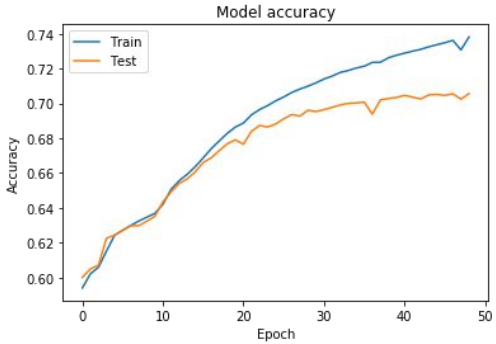


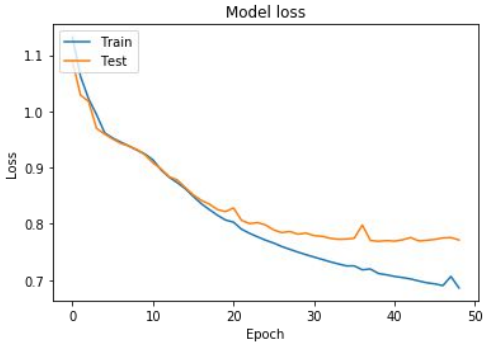
Rede Neural para *mod*



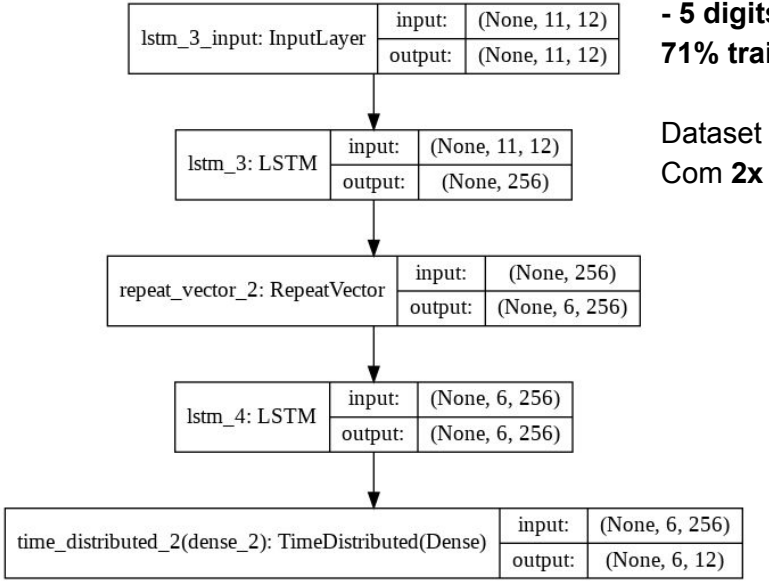
- 5 digits reversed: + 1 layer LSTM (128 HN), 200k training examples = 70% train/test accuracy in 49 epochs

Dataset ajustado para casos onde o resultado é diferente dos operandos.



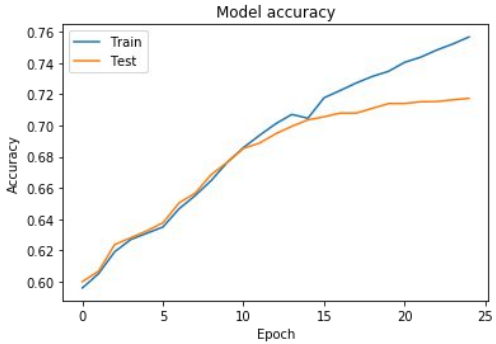


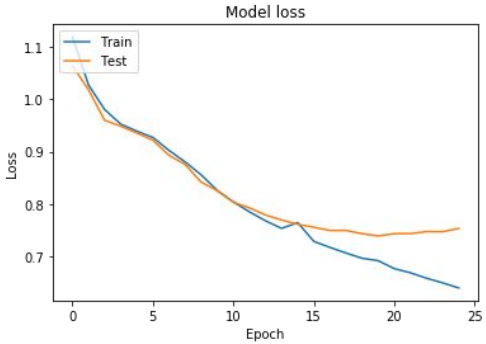
Rede Neural para *mod*



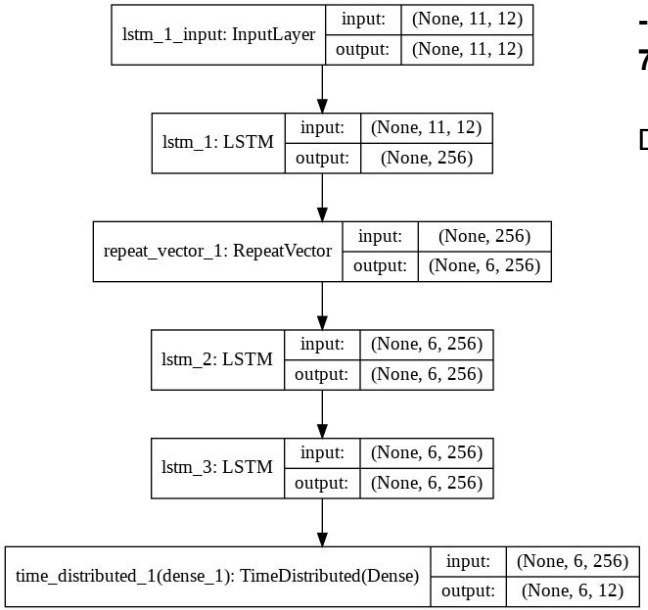
- 5 digits reversed: + 1 layer LSTM (256 HN), 200k training examples = 71% train/test accuracy in 25 epochs

Dataset ajustado para casos onde o resultado é diferente dos operandos.
Com **2x (256)** memory cells



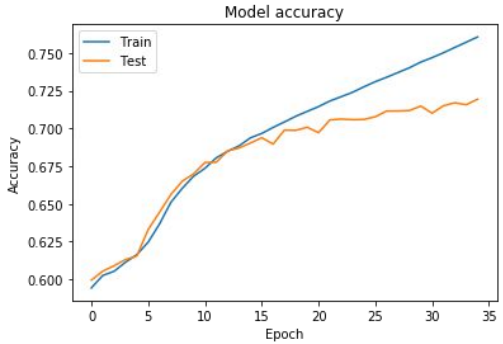


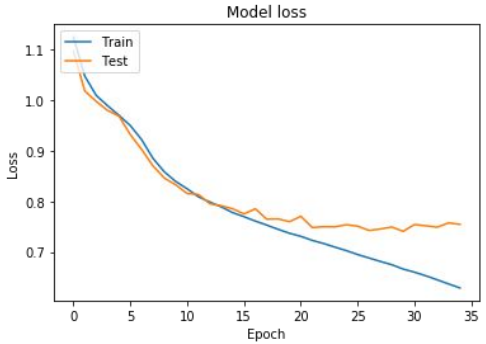
Rede Neural para *mod*



- 5 digits reversed: + 2 layers LSTM (128 HN), 200k training examples = 71% train/test accuracy in 35 epochs

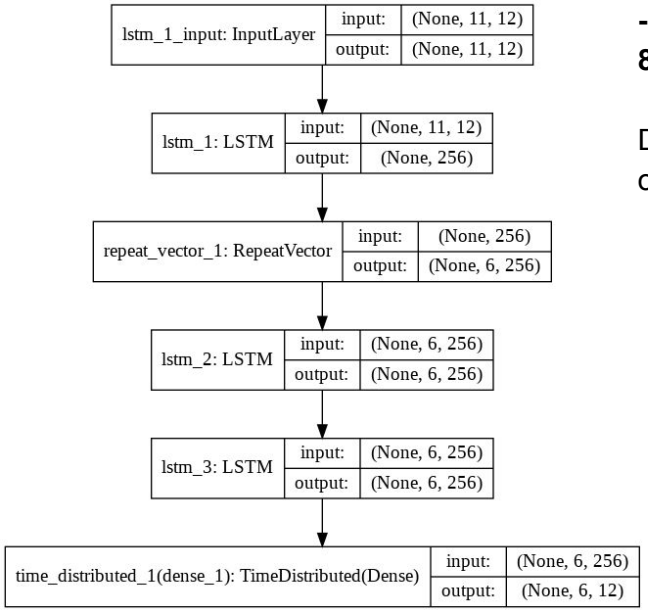
Dataset ajustado para casos onde o resultado é diferente dos operandos.





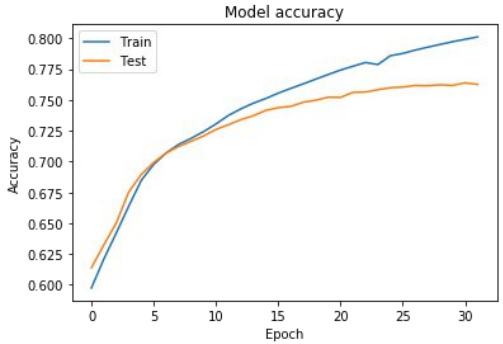
COPPE UFRJ
Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa de Engenharia

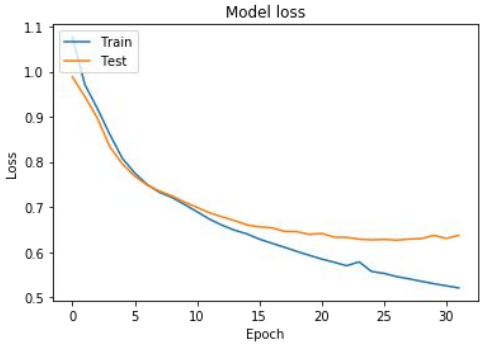
Rede Neural para *mod*



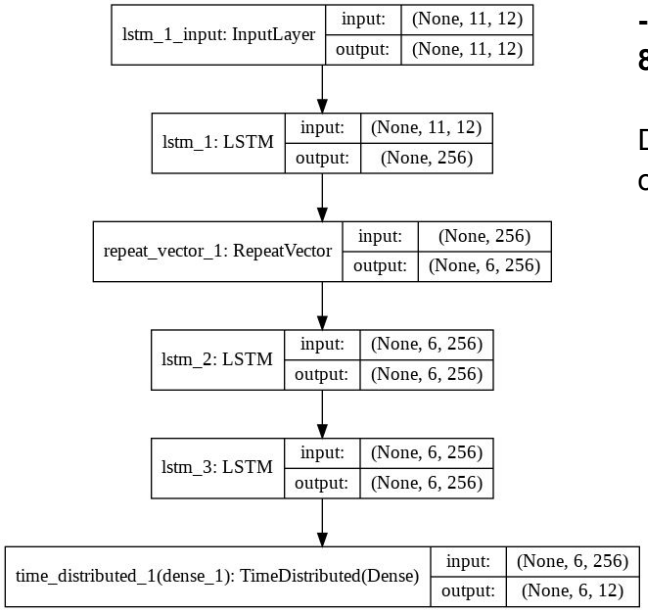
- 5 digits reversed: + 1 layer LSTM (256 HN), 500k training examples = 80/76 % train/test accuracy in 32 epochs

Dataset (maior) ajustado para casos onde o resultado é diferente dos operandos.



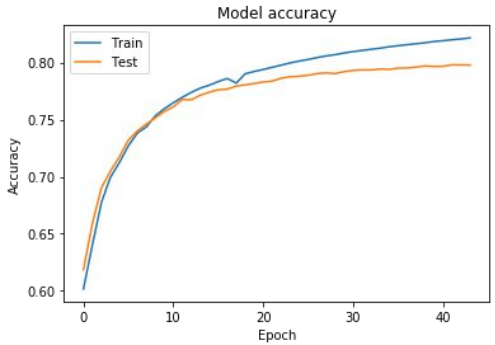


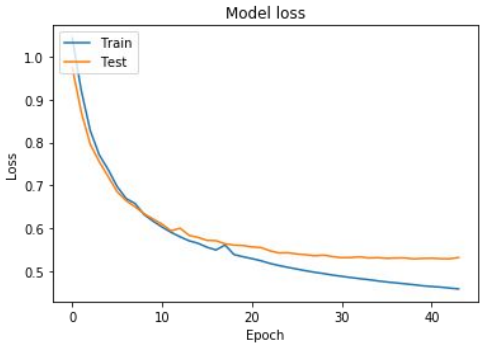
Rede Neural para *mod*



- 5 digits reversed: + 1 layer LSTM (256 HN), 1M training examples = 80 % train/test accuracy in 44 epochs

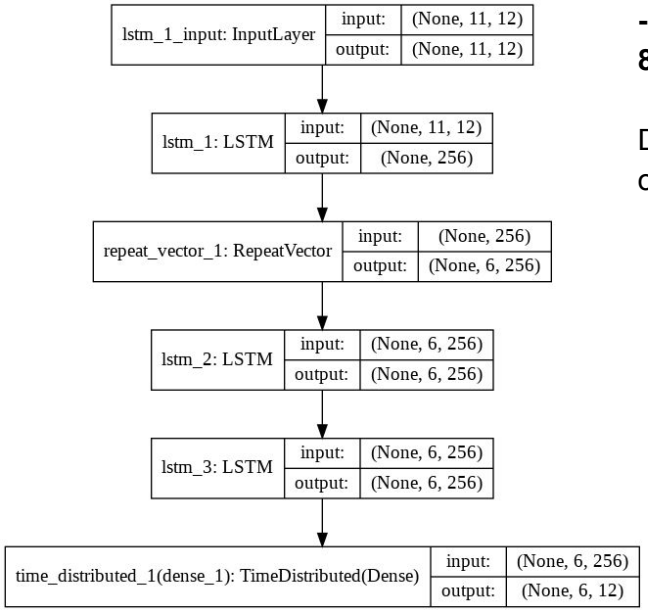
Dataset (bem maior) ajustado para casos onde o resultado é diferente dos operandos.





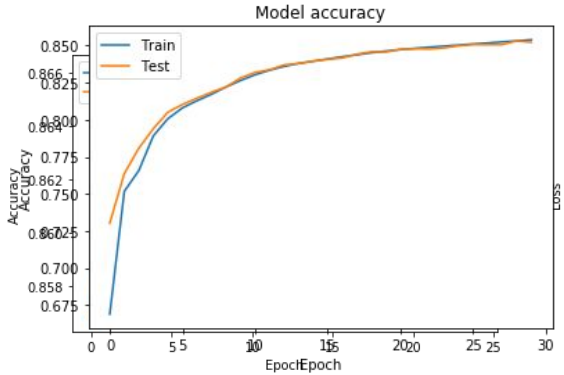
COPPE UFRJ
Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa de Engenharia

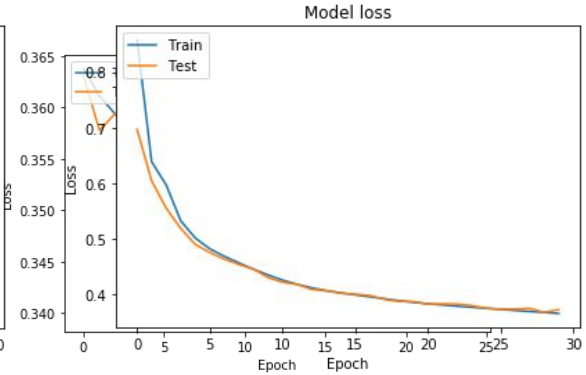
Rede Neural para *mod*



- 5 digits reversed: + 1 layer LSTM (256 HN), 10M training examples = 85 % train/test accuracy in 29 epochs

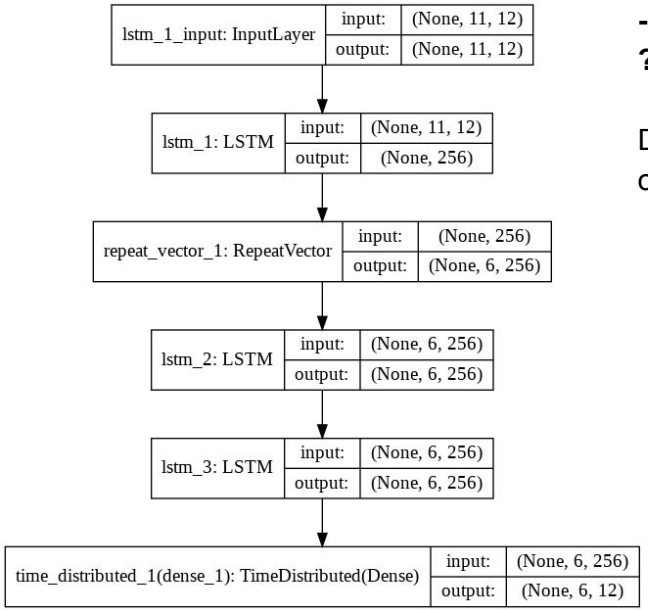
Dataset (bem maior) ajustado para casos onde o resultado é diferente dos operandos.





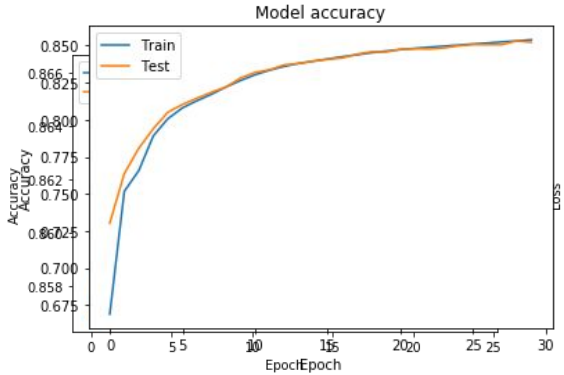
COPPE UFRJ
Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa de Engenharia

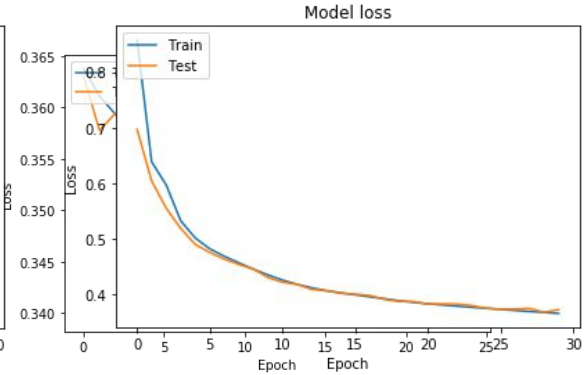
Rede Neural para *mod*



- 5 digits reversed: + 1 layer LSTM (256 HN), 100M training examples = ?? % train/test accuracy in ?? epochs

Dataset (bem maior) ajustado para casos onde o resultado é diferente dos operandos.

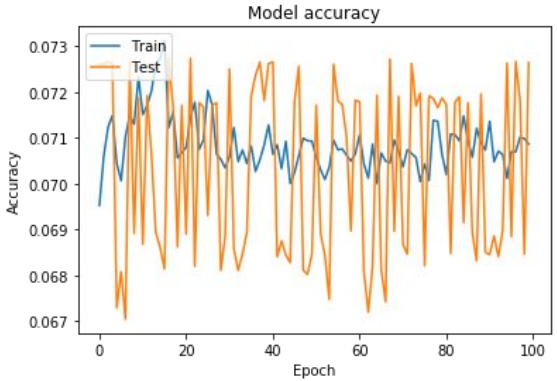


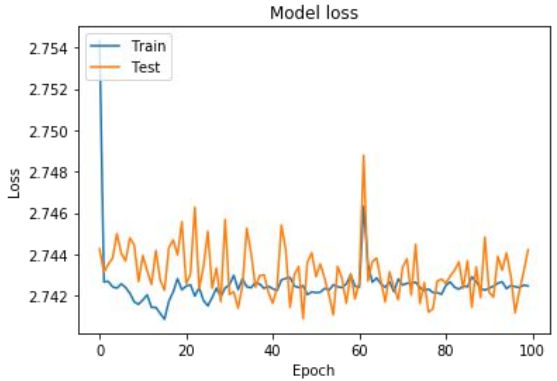


Conclusões

- Operações de **adição** conseguem ser cobertas com redes LSTM.
- Operações de **mod** ainda precisam ser melhor estudadas.
  - Abordagens diretas não funcionam, é preciso separar o problema.

64 digits, 1 layer LSTM (256 HN), 100k training examples = 7% train/test accuracy in 100 epochs



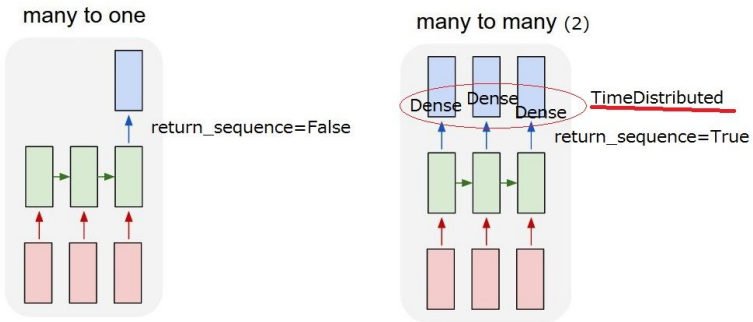


Conclusões

- Operações de **adição** conseguem ser cobertas com redes LSTM.
- Operações de **mod** ainda precisam ser melhor estudadas. Opções:
  - Abordagens diretas não funcionam, é preciso separar o problema.
  - Menor tamanho de batch (aumenta bastante o tempo de treinamento), sem resultados evidentes
  - Datasets maiores linearmente (100k,200k,500k não apresentaram mudanças significativas)
  - Datasets maiores em ordem de grandeza sim! (1M, 10M,...)
- Google Colab não funciona muito bem com LSTM (PaperSpace).

Próximos passos

- Operações de **mod**:
  - *return_sequences=True*



https://stackoverflow.com/questions/42755820/how-to-use-return-sequences-option-and-timedistributed-layer-in-keras

Próximos passos

- Operações de **mod**:
  - `return_sequences=True`
  - Utilizar outros tipos de redes (baseadas em mecanismos de atenção)
  - Utilizar uma topologia diferente