A r q u i t e t u r a T r a n s f o r m e r

PyData

luiz.andrade@tevec.com.br

APRESENTAÇÃO

Engenheiro civil pela Escola Politécnica da USP

Msc em Engenharia de Sistemas Logísticos pela Escola Politécnica da USP

(*in course*) PhD em Engenharia de Transportes pela Escola Politécnica da USP

M.B.A. em Global Supply Chain e Logistics pelo Massachussets Institute of Technology

12 anos de experiência com empreendedorismo e tecnologia. Fundação de duas start-ups ambas em estágio pré Série A

Professor voluntário da School of AI São Paulo

luiz.andrade@tevec.com.br

TIPOS DE INFERÊNCIA

DEDUÇÃO

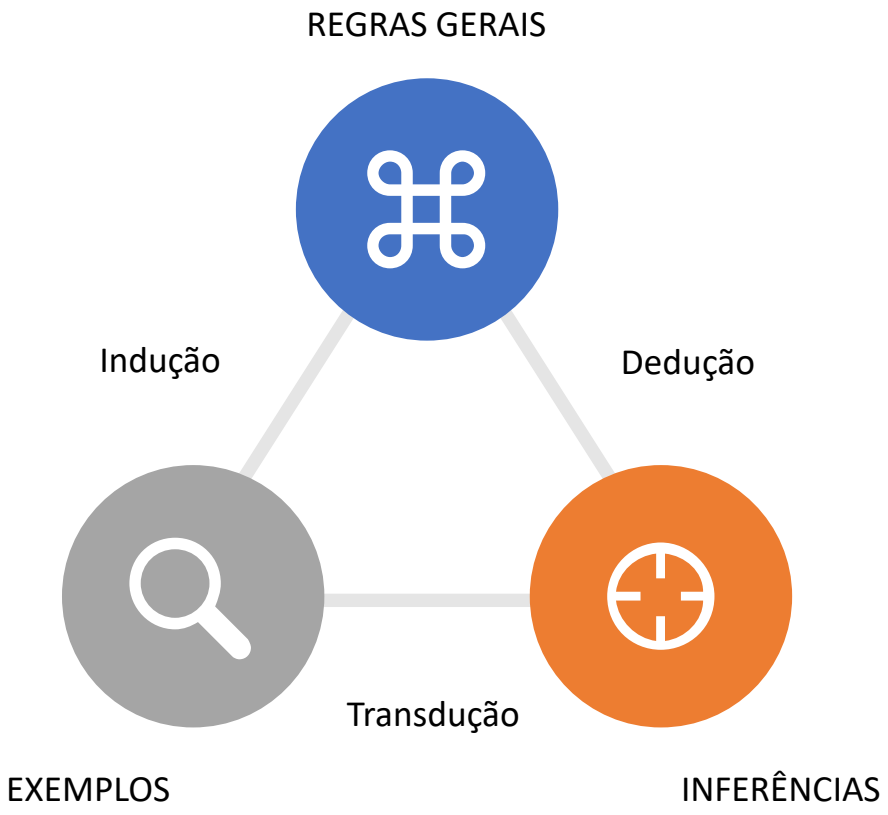
A dedução é um método de inferência que, através da análise de regras gerais fazer afirmações sobre uma observação particular. Exemplos do método dedutivo são: teoremas, códigos gramaticais, códigos jurídicos.

INDUÇÃO

Através da observação de exemplos, tenta-se com a indução modelar regras mais gerais. Busca-se modelar o fenômeno gerador. O método indutivo é a base da teoria do aprendizado estatístico e aprendizado de máquina

TRANSDUÇÃO

Em casos em que a regra geral é muito complexa, e/ou a quantidade de exemplos é insuficiente para chegar-se a uma descrição satisfatória do fenômeno Gerado, utilizam-se exemplos conhecidos para fazer afirmações sobre outros exemplos desconhecidos.



luiz.andrade@tevec.com.br

COMO APRENDEMOS A FALAR?

A linguagem é um caso perfeito para aplicar conceitos de transdução. Nós aprendemos a falar muito **antes de saber o código gramatical** da nossa lingua. Mesmo antes de sabermos quais são as regras gerais, **através da repetição de padrões** conseguimos gerar **exemplos de frases corretas**. Conseguimos reproduzir exemplos sem conhecer a regra geral. Na maioria das vezes, as pessoas não usam a regra gramatical para se comunicar.

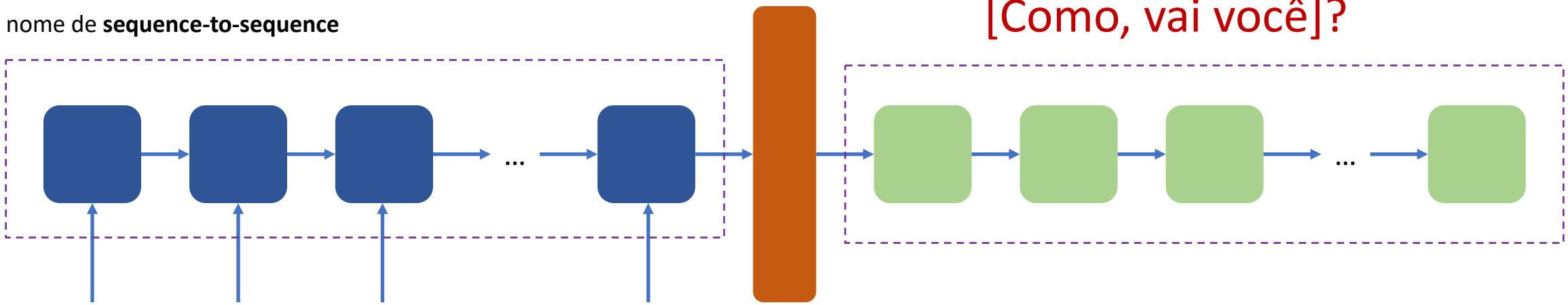
VS.

luiz.andrade@tevec.com.br

ENCODER-DECODER

Em geral, em problemas de NLP o encoder e o decoder são redes neurais recorrentes e o modelo assume o nome de **sequence-to-sequence**

Estado codificado representando a informação comprimida de uma sequencia



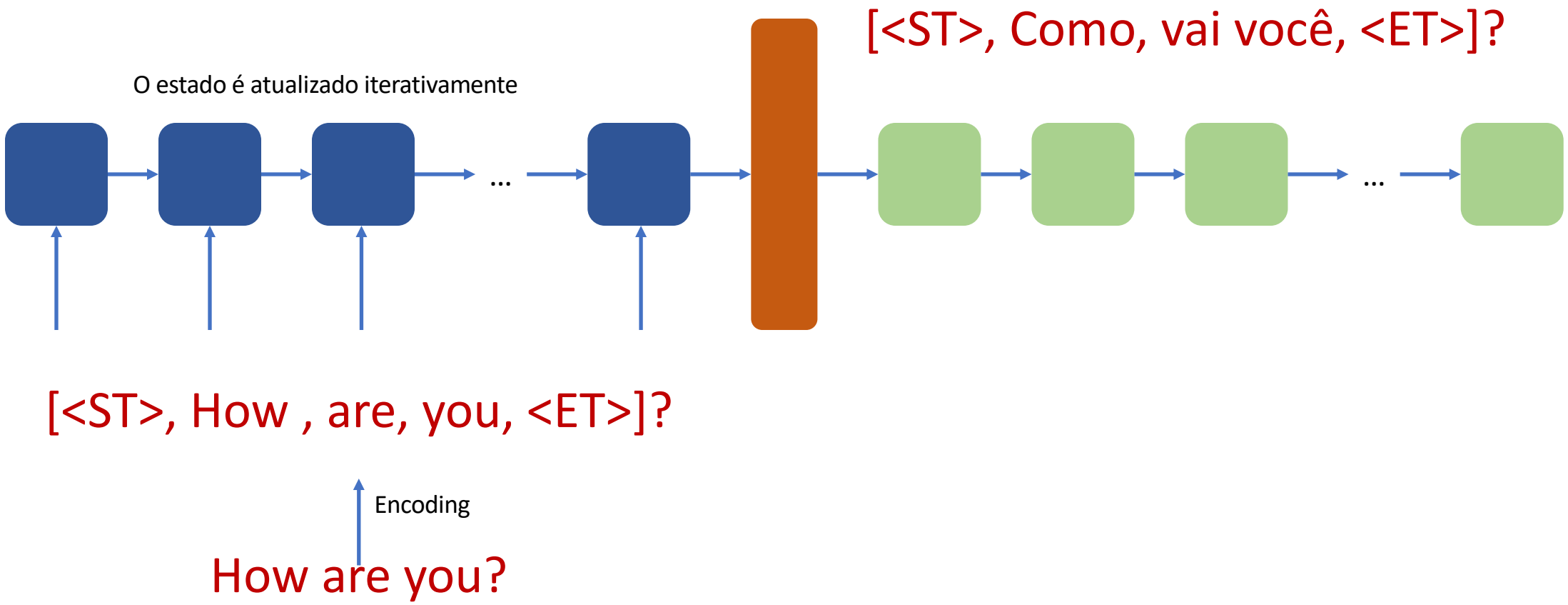
[How , are, you]?

Encoding

How are you?

luiz.andrade@tevec.com.br

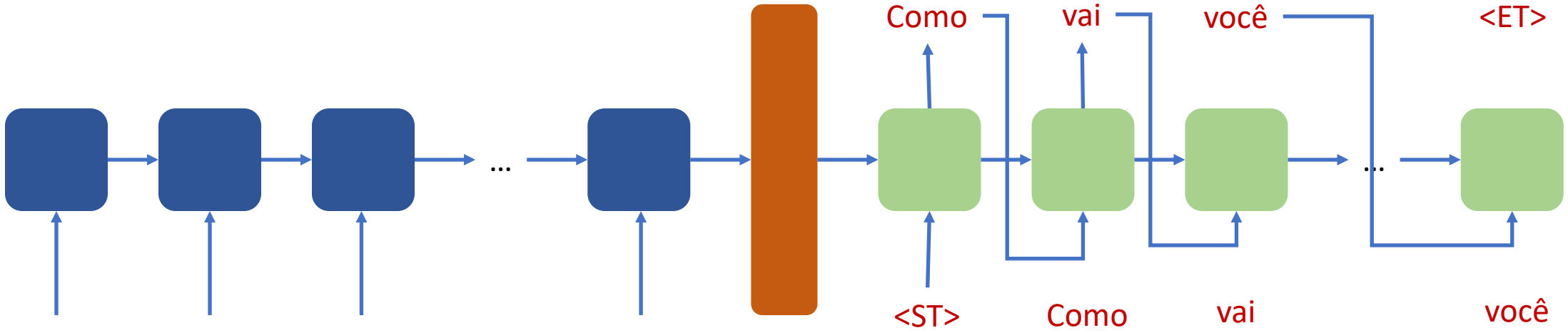
ENCODER-DECODER



luiz.andrade@tevec.com.br

ENCODER-DECODER

O treinamento é feito por um processo chamado teacher-forcing



[<ST>, How , are, you, <ET>]?

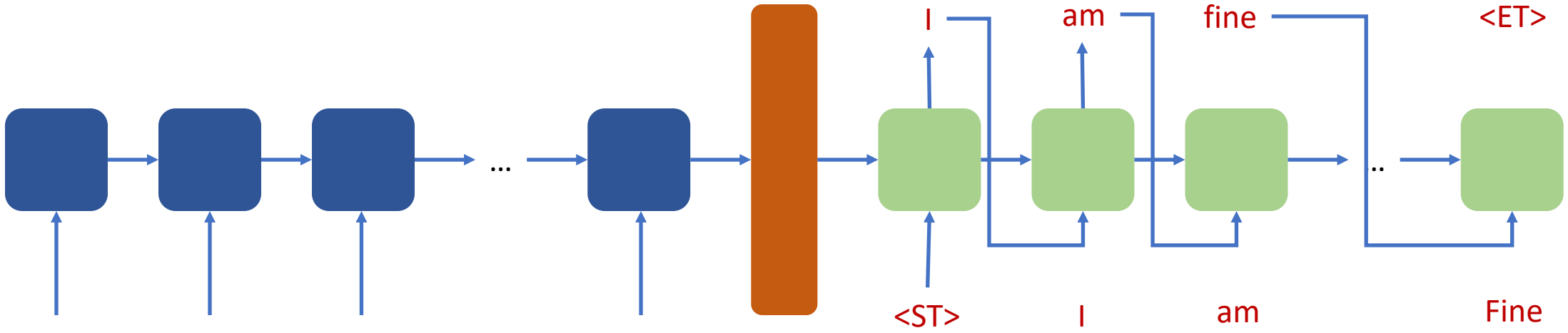
Encoding

How are you?

luiz.andrade@tevec.com.br

ENCODER-DECODER

Podemos trocar os outputs e transformar o problema em Q&A



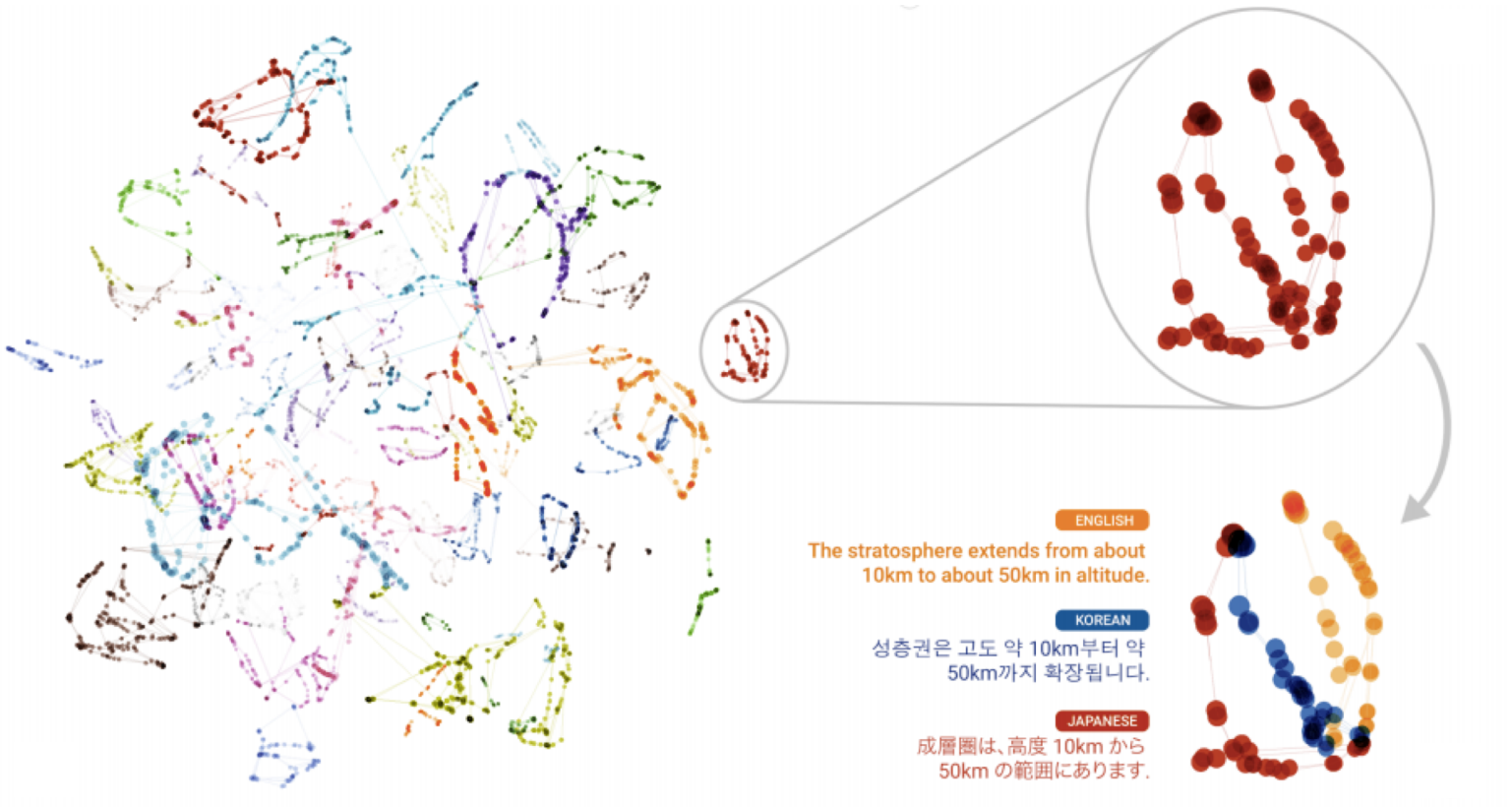
[<ST>, How , are, you, <ET>]?

Encoding

How are you?

luiz.Andrade@tevec.com.br

ESPAÇO CODIFICADO



Na imagem, cada cluster (cor) representa uma frase de um total de 74 frases traduzidas pelo algoritmo em 3 linguas diferentes (inglês, koreano e japones).

Podemos perceber que uma mesma frase, escrita em línguas diferentes forma clusters compactos no espaço latente. Pode-se inclusive perceber que a “trajetória” de cada frase em linguas diferentes é similar.

Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F., Wattenberg, M., Corrado, G., Hughes, M., Dean, J. (2016) Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. arXiv:1611.04558

luiz.andrade@tevec.com.br

ENCODER-DECODER COM RNN'S

APRESENTAM BONS RESULTADOS

Muitos dos benchmarks de problemas de NLP ainda não detidos por modelos sequence-to-sequence. Desde sua primeira proposção, diversas arquiteturas vem sendo propostas com melhorias de desempenho computacional e precisão.

RNN'S SÃO UMA FORMA "NATURAL" DE TRATAR SEQUENCIAS

A natureza sequencial da RNN's se adequa bem a problemas de processamento de sequencias. Tensores de frases são facilmente codificados para serem processados por modelos sequence-to-sequence. Com isso há pouca necessidade de *feature engineering*.

ALTO CUSTO COMPUTACIONAL

Como o estado é calculado iterativamente, considerando cada elemendo do vetor de inputs, as RNNs são difíceis de terem seu treinamento paralelizado. Além disso, alguns tipos de RNNs sofre do problema de *vanishing/exploding gradient*.

MEMÓRIA DE CURTO PRAZO

Para sequencias de longa duração, o erro de inferência é retroprogadado muitas vezes nos estados internos da RNN. Mesmo com o processo de otimização de pesos corretamente ajustado, isso faz com que os gradientes fiquem excessivamente pequenos, tornando difícil a propagação de efeitos de longo prazo na sequencia.

NÃO CONSIDERA TODO O CONTEXTO

Por processar cada input sequencialmente, o próximo valor de output só pode depender de inputs anteriores. Na realidade a interpretação de textos, tradução e Q&A não funcionam dessa forma. O modelo poderia interpreter toda a sequencia para dar uma resposta melhor.

luiz.andrade@tevec.com.br

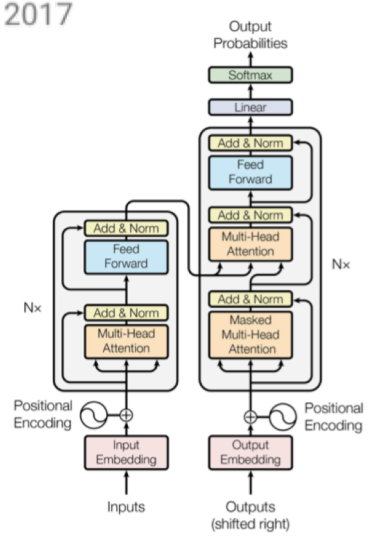
TRANSFORMER (2017)

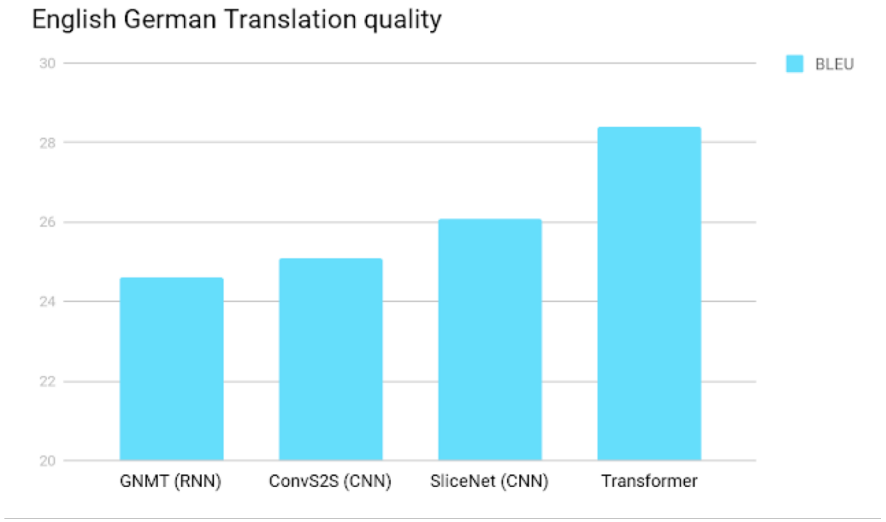
Google AI Blog

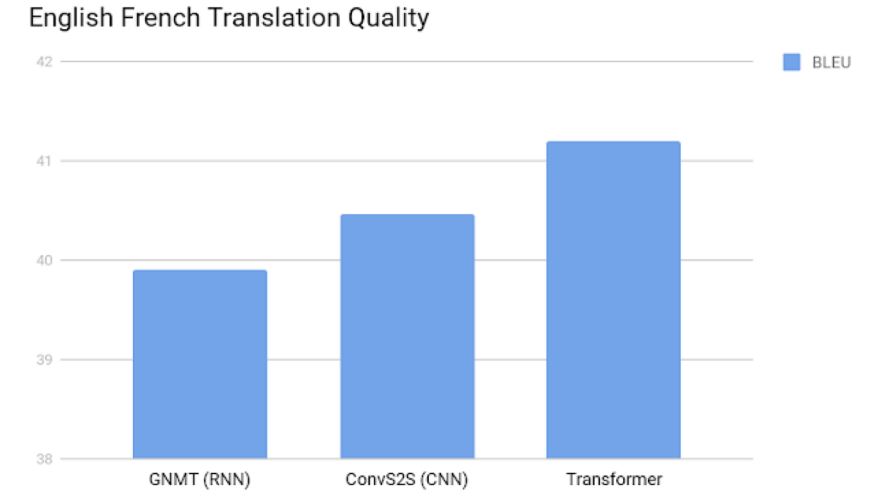
The latest news from Google AI

Transformer: A Novel Neural Network Architecture for Language Understanding

Thursday, August 31, 2017



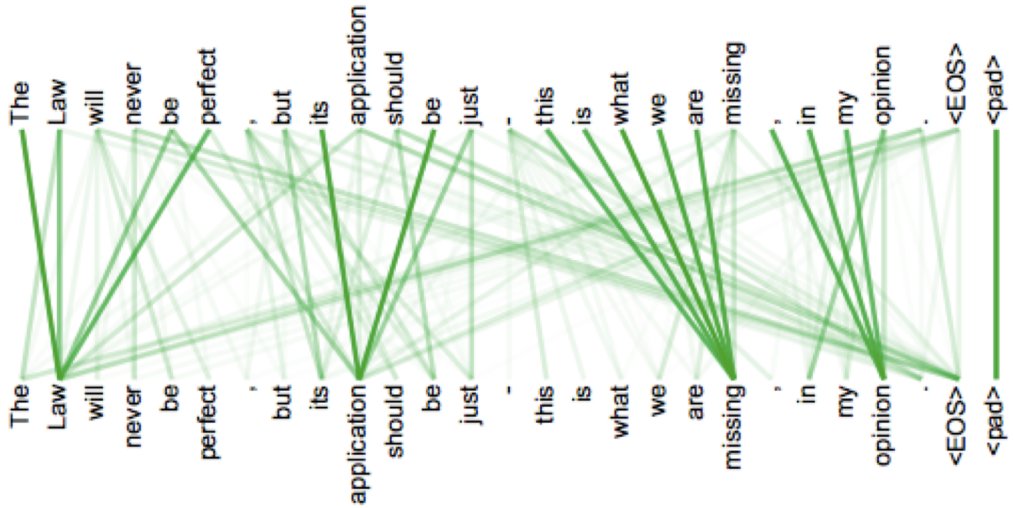


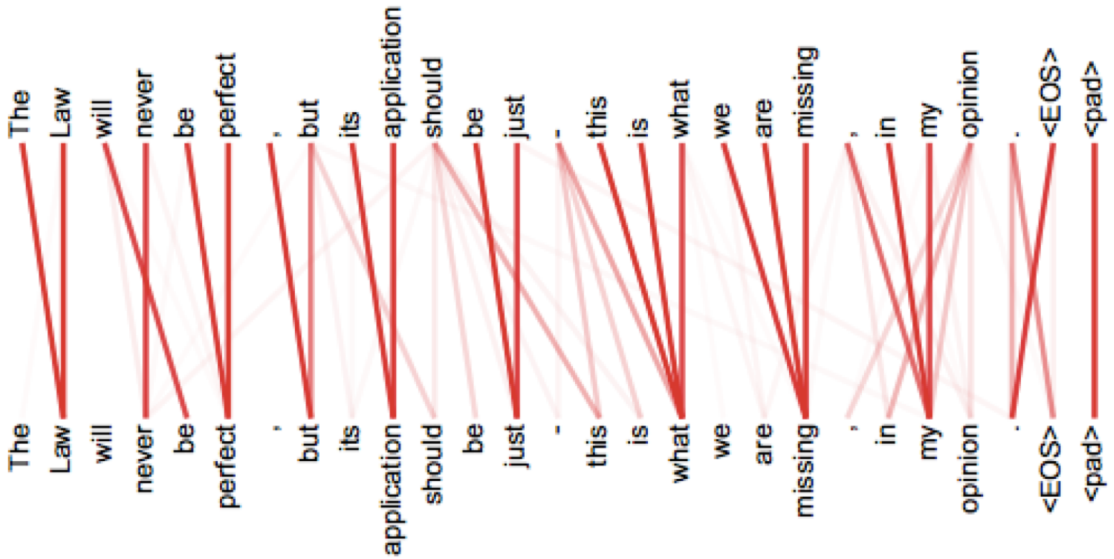


luiz.andrade@tevec.com.br

INTUIÇÃO DOS MECANISMOS DE ATENÇÃO

Os mecanismos de atenção nos ajuda a visualizer o que o modelo está relacionando quando está processando um input. Isso nos dá uma visão do relacionamento entre as palavras. Assim como uma CNN “aprende” uma quantidade de kernels e mapeia features em uma imagem, uma rede com mecanismos de atenção “aprende” o significado entre posições relativas de paravras.

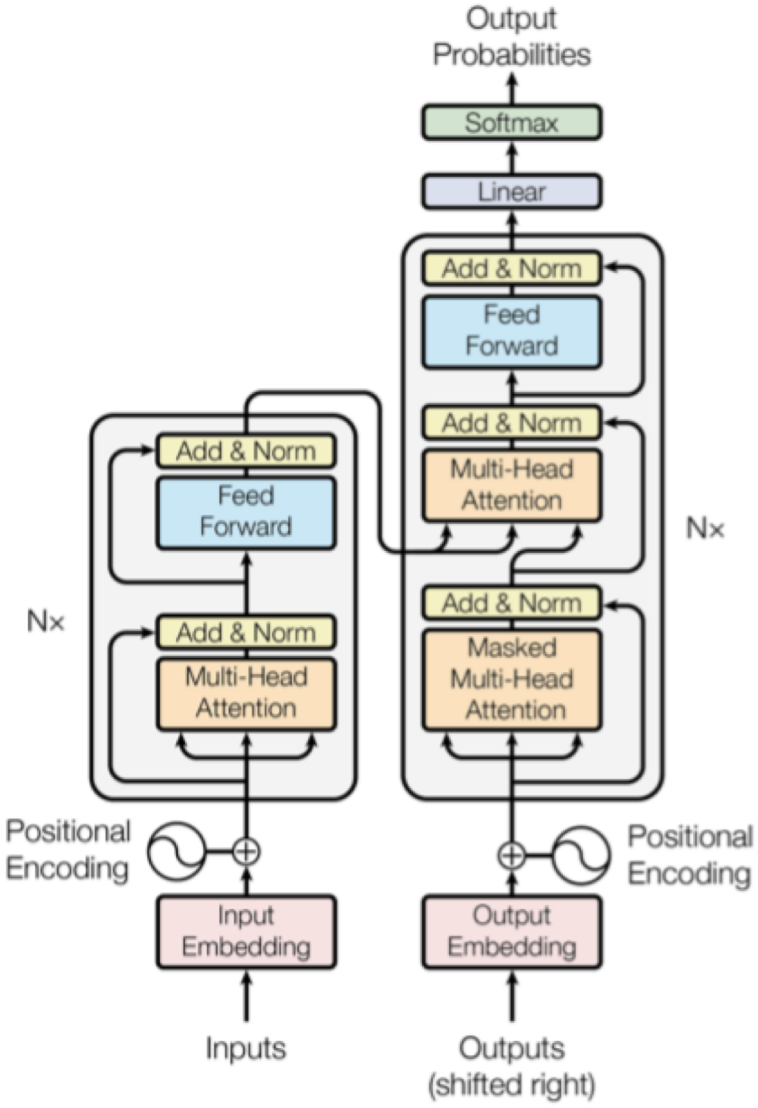


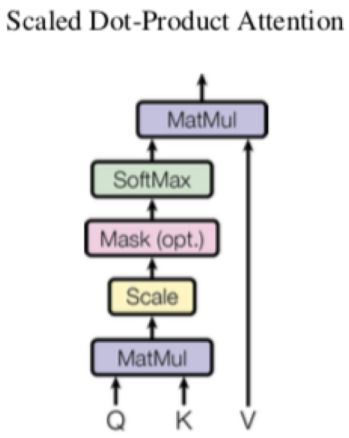


Vaswani A., Shazeer, N., Parmer N. et Al. (2017) Attention is all you need. NIPS'17 Proceedings of the 31st International Conference on Neural Information Processing Systems

luiz.andrade@tevec.com.br

TRANSFORMER

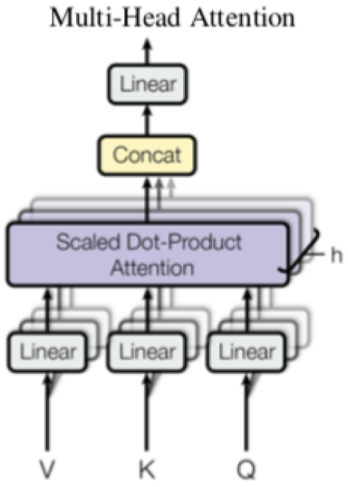




Q = Queries ~ Palavras de Entrada

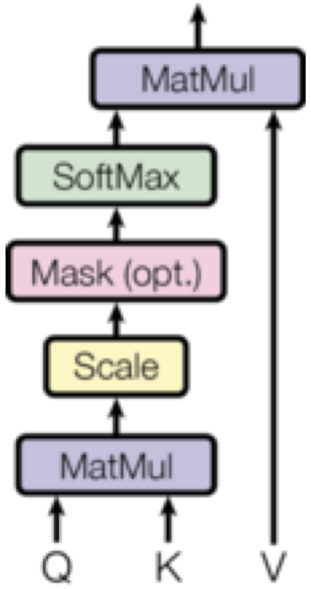
K = Keys ~ Palavras Afins

V = Values ~ Valores



luiz.andrade@tevec.com.br

TRANSFORMER

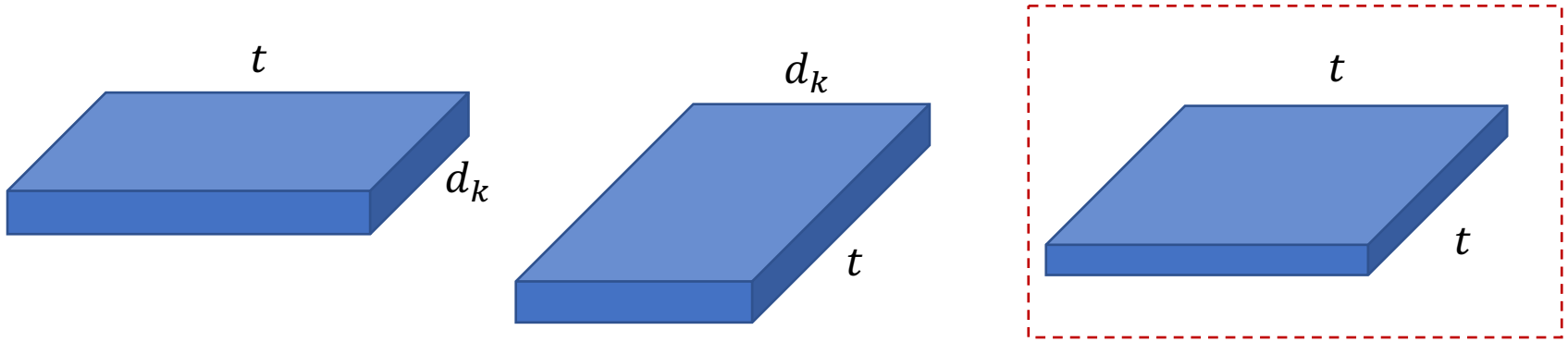


O modelo utiliza uma forma de atenção multiplicativa.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Visualizando:

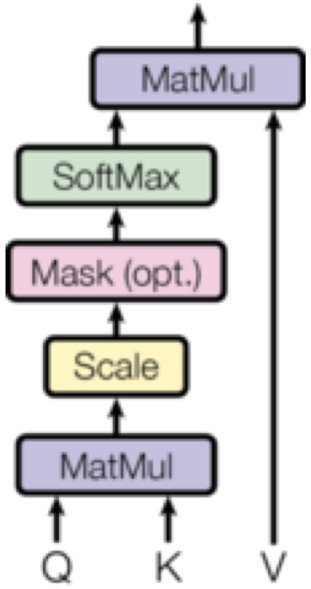
$$q \times k^T = qk^T$$



Mapa de atenção

luiz.andrade@tevec.com.br

TRANSFORMER

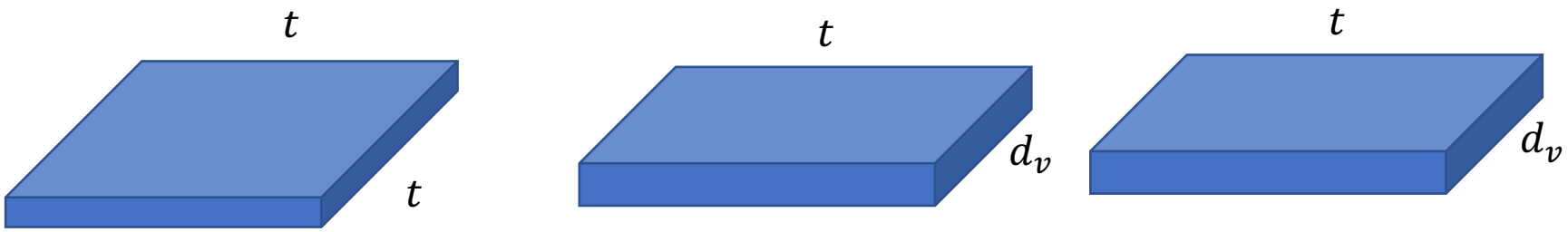


O modelo utiliza uma forma de atenção multiplicativa.

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

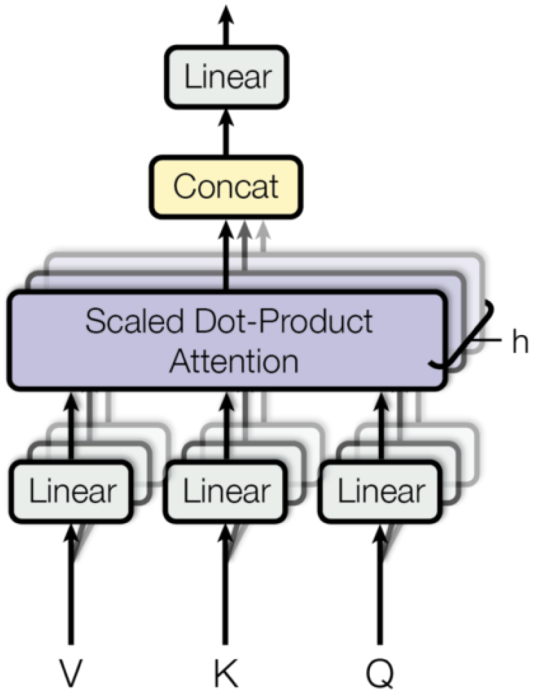
Visualizando:

$$qk^T \times v = (qk^T)v$$



luiz.andrade@tevec.com.br

TRANSFORMER

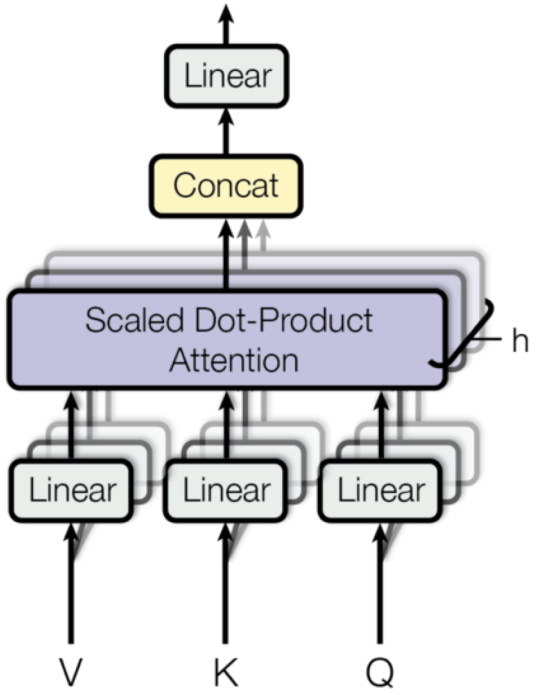


O modelo utiliza várias projeções lineares de Q, K e V para gerar múltiplas instâncias de atenção. Isso é similar aos feature maps de camadas convolucionais

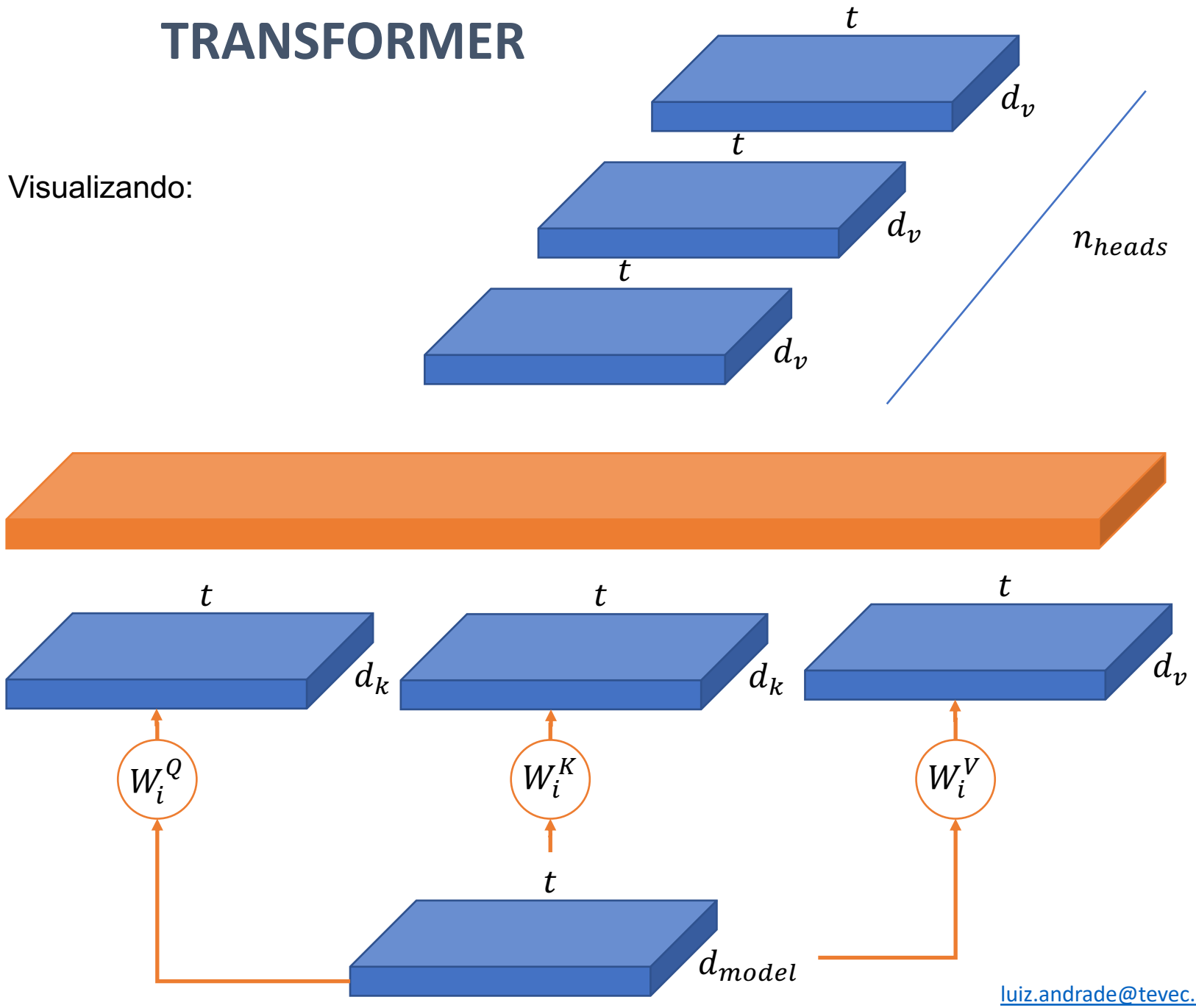
$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head_1}, ..., \mathrm{head_h})W^O$$
$$\text{where } \mathrm{head_i} = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

luiz.andrade@tevec.com.br

TRANSFORMER

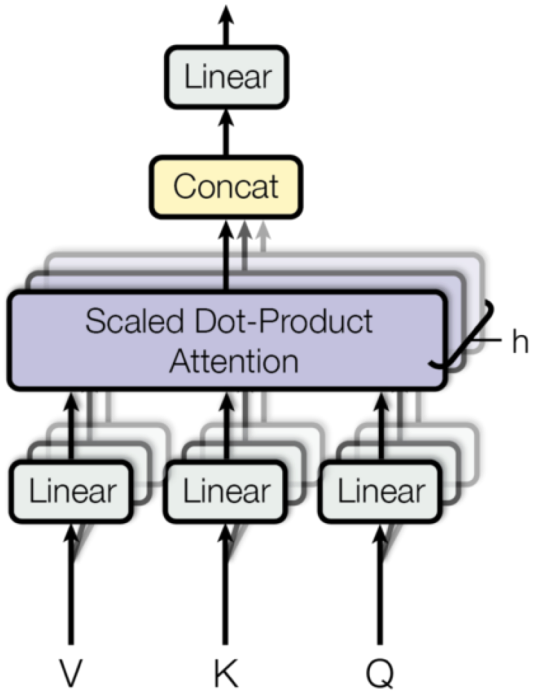


Visualizando:

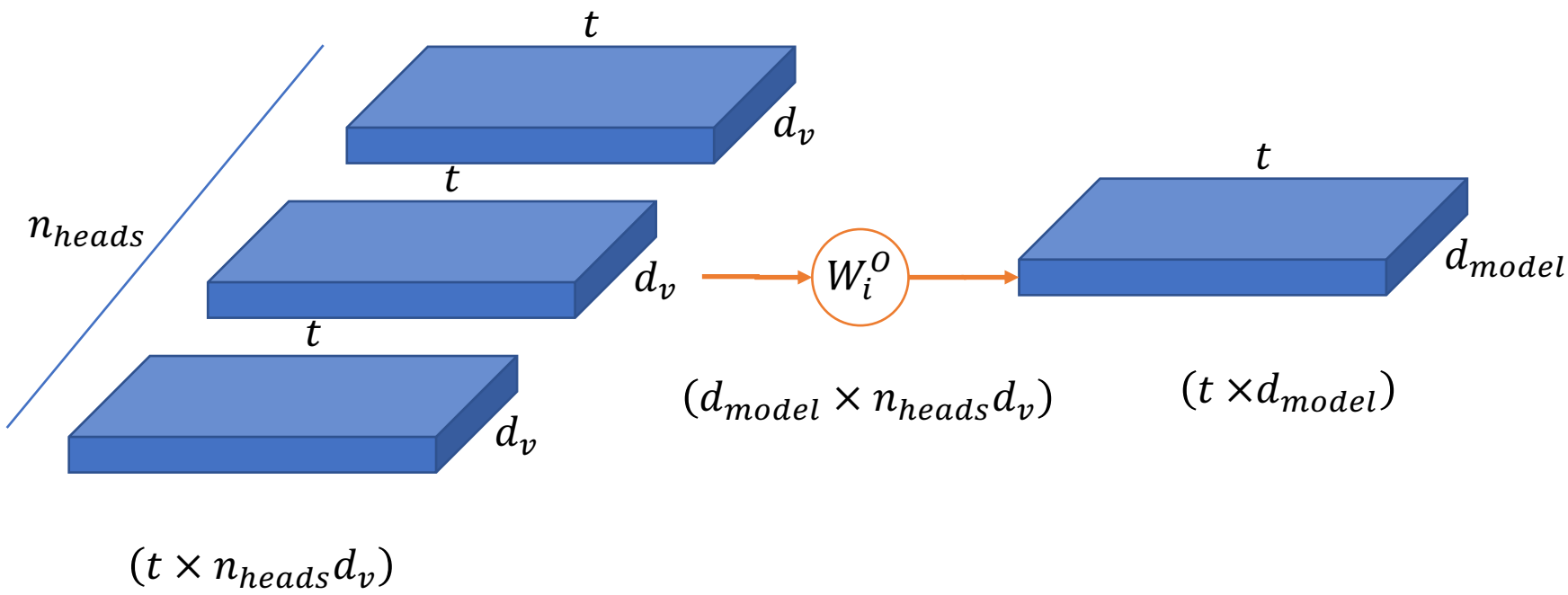


luiz.andrade@tevec.com.br

TRANSFORMER

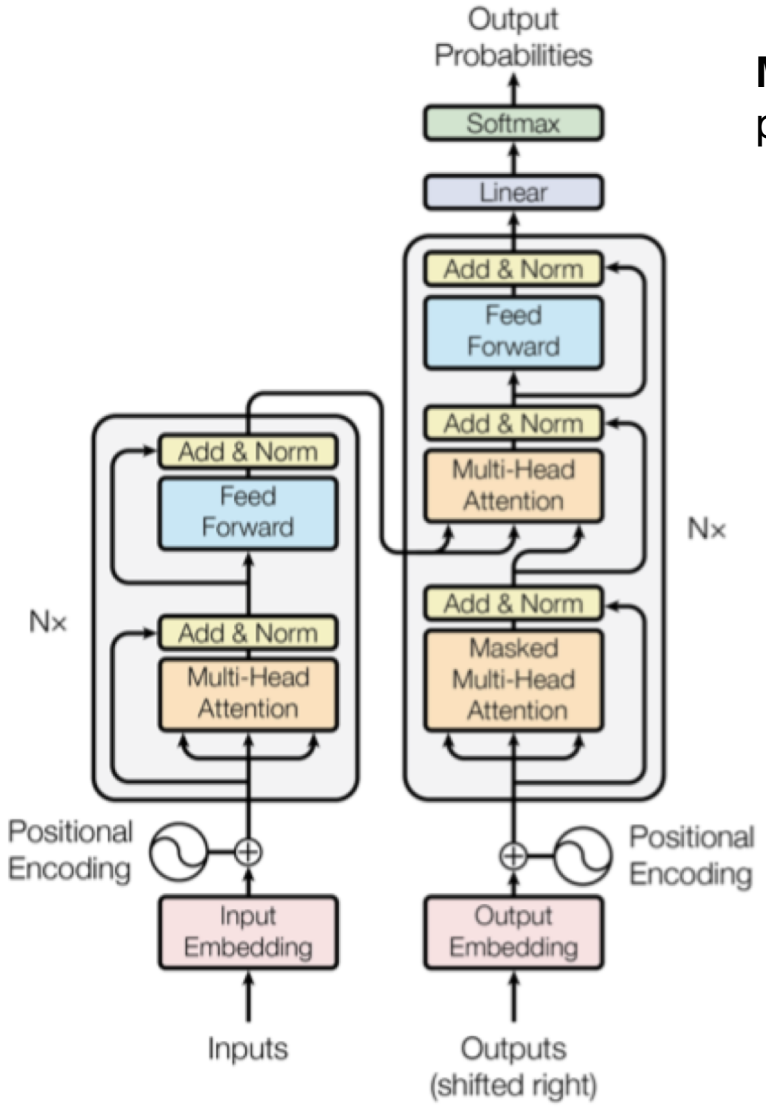


Visualizando:

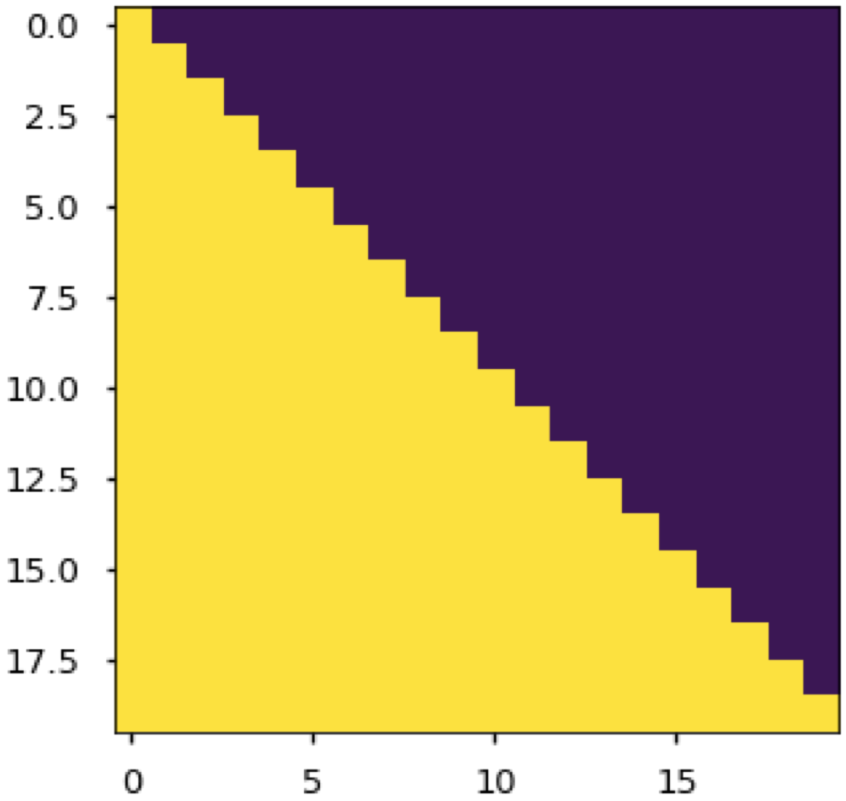


luiz.andrade@tevec.com.br

TRANSFORMER



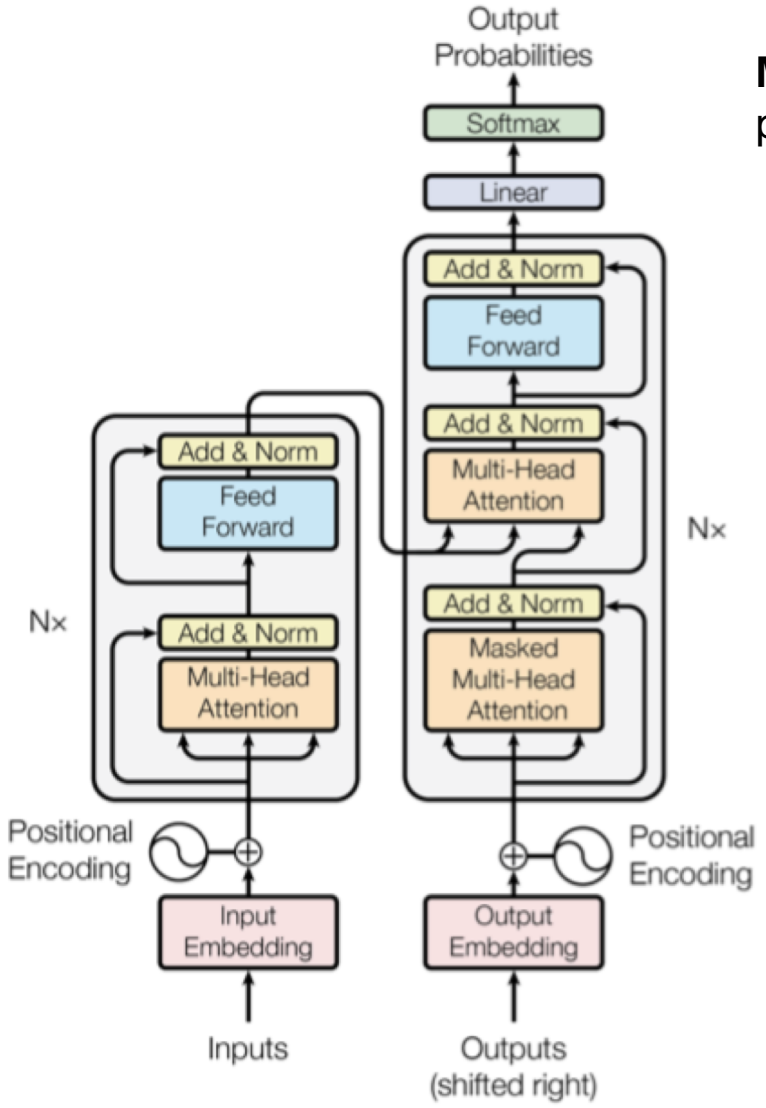
Masking: forma de impeder que informação do “futuro” ajude a prever o passado.



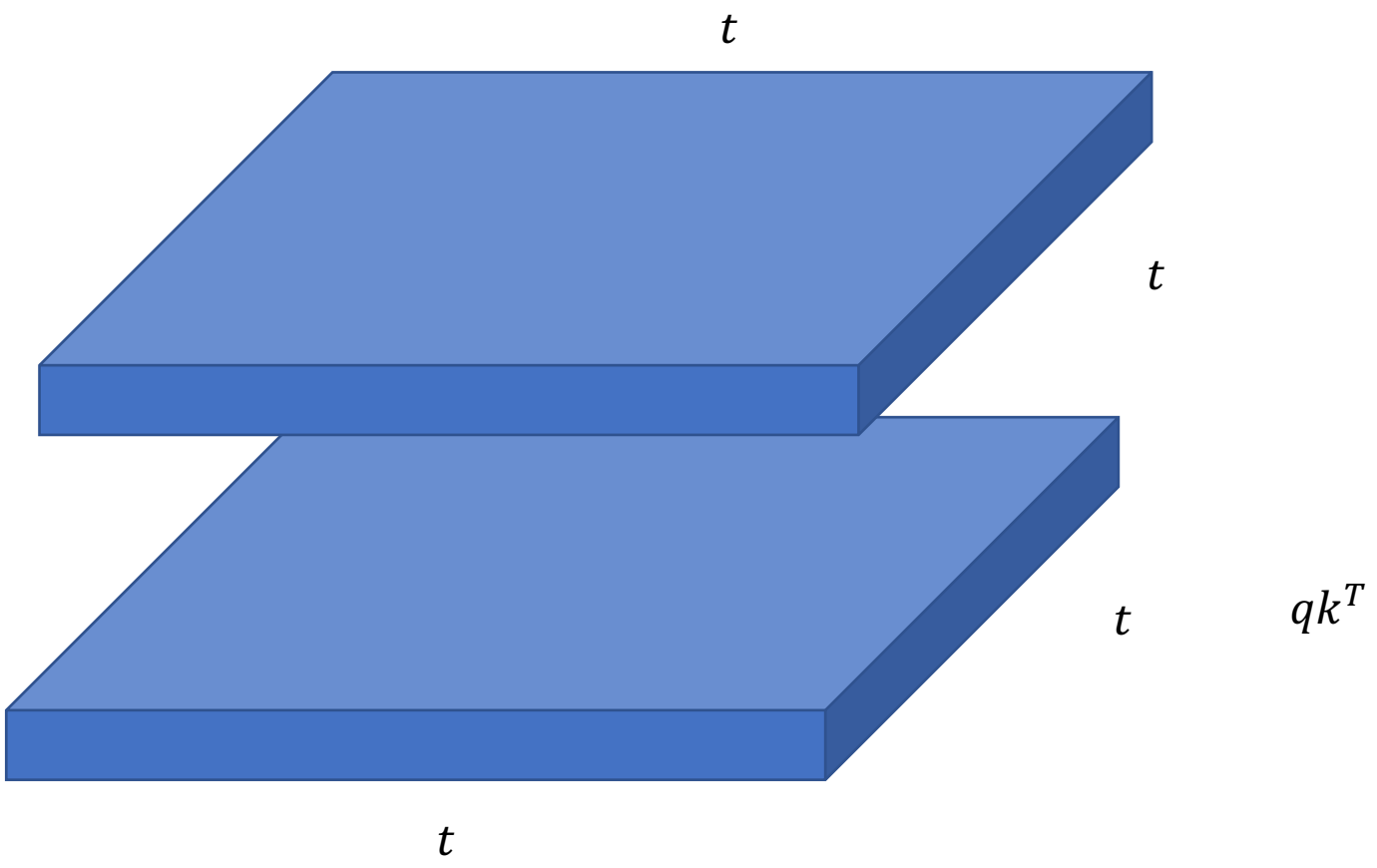
http://nlp.seas.harvard.edu/2018/04/03/attention.html

luiz.andrade@tevec.com.br

TRANSFORMER



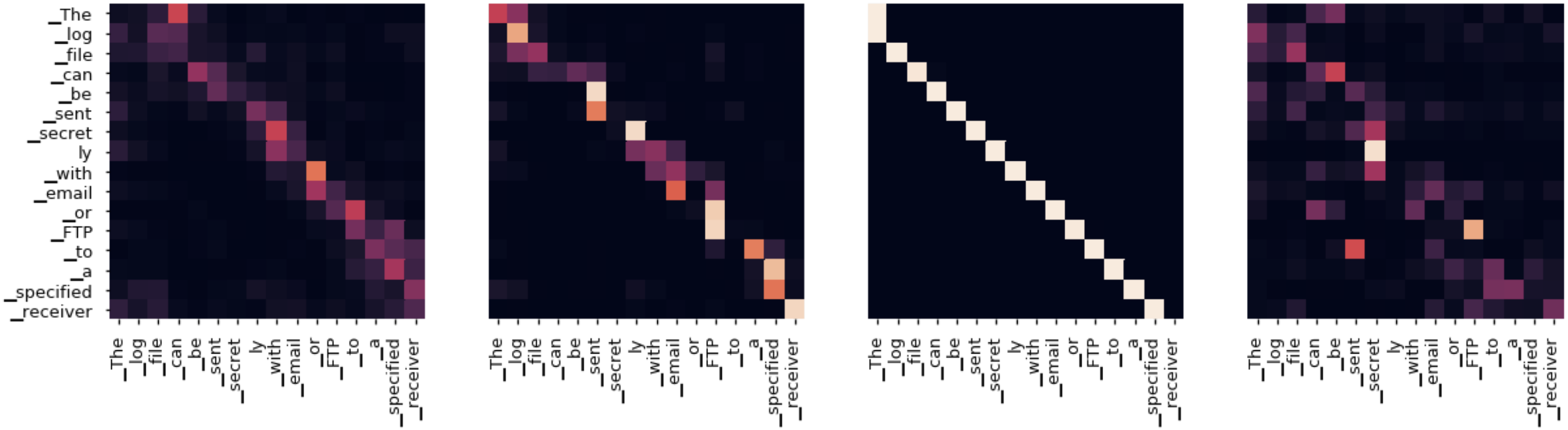
Masking: forma de impeder que informação do “futuro” ajude a prever o passado.



luiz.andrade@tevec.com.br

TRANSFORMER

Visualização de uma camada de atenção do **encoder** de um problema de tradução

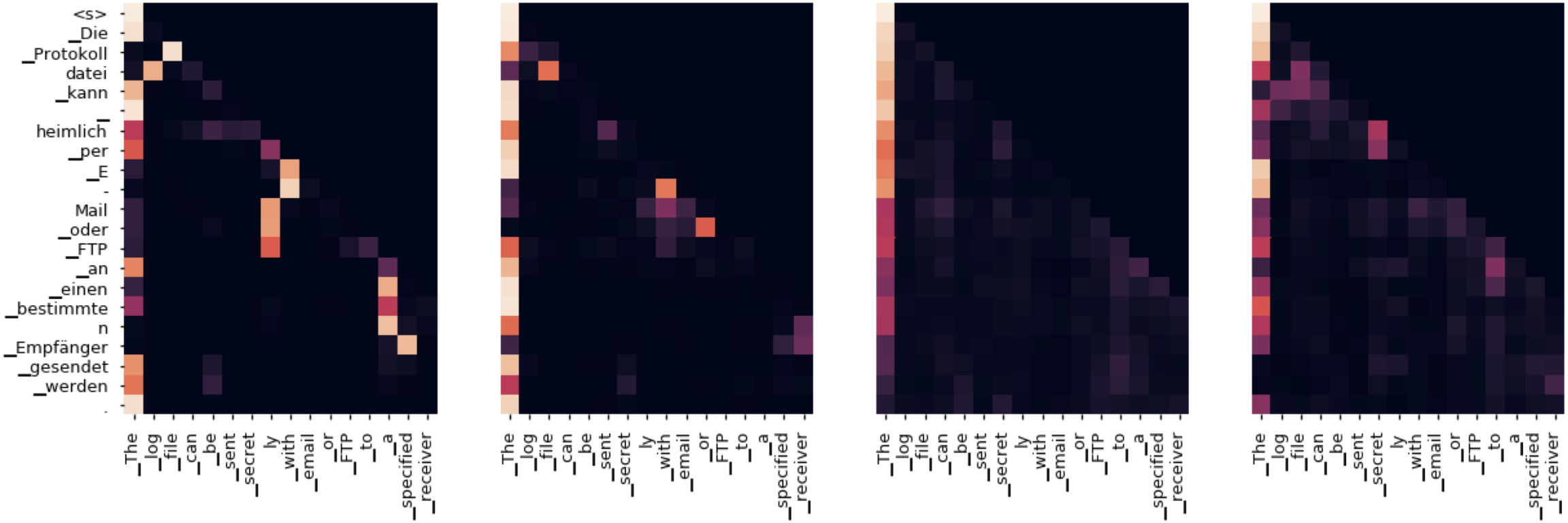


http://nlp.seas.harvard.edu/2018/04/03/attention.html

luiz.andrade@tevec.com.br

TRANSFORMER

Visualização de uma camada de atenção do **decoder** de um problema de tradução



http://nlp.seas.harvard.edu/2018/04/03/attention.html

luiz.andrade@tevec.com.br

TRANSFORMER

O TREINAMENTO PODE SER PARALELIZADO

Uma vez que o transformer se baseia exclusivamente em mecanismos de atenção, não há necessiade de processamento em série dos inputs. Com isso o treinamento pode ser paralelizado.

ANÁLISE VISUAL DOS MECANISMOS DE ATENÇÃO

Se comparados a células de RNN mais convencionais (Vanila RNN, LSTM e GRU), os mecanismos de atenção permitem uma melhor interpretação das features que o modelo aprende. Isso se dá pela inspeção direta dos scores de atenção.

APRESENTAM BONS RESULTADOS

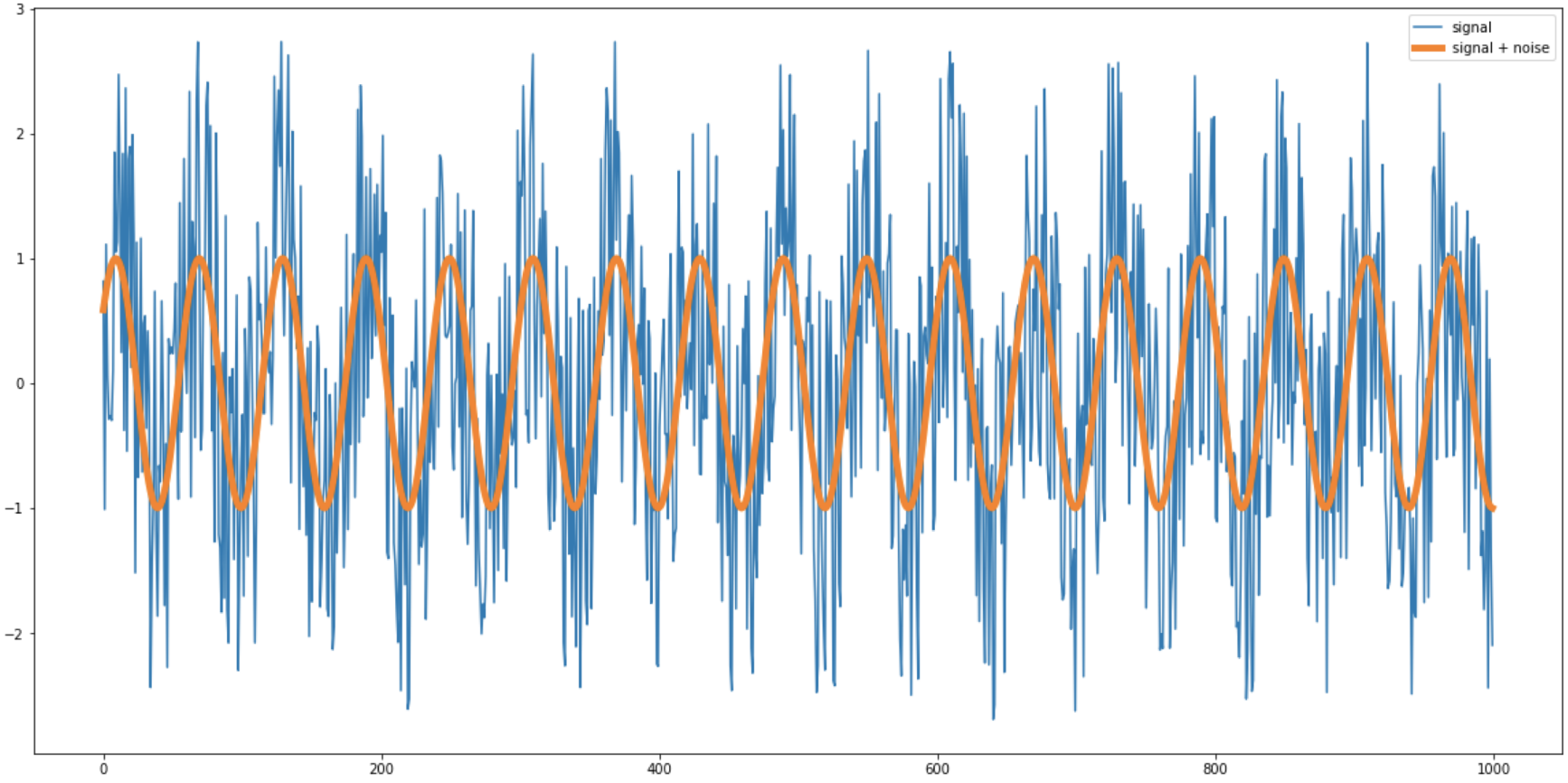
Para o problema de tradução de linguagem (machine translation) quando proposto, o modelo bateu todos os benchmarks. Para problemas associados de Q&A e text summarization o modelo tamb'm apresenta excelentes resultados.

luiz.andrade@tevec.com.br

EXPERIMENTOS COM SÉRIES TEMPORAIS

Experimento 1 – DeNoiser

Dada uma série de tempo, formada de um sinal e um ruido, o DeNoiser deve identificar o sinal e separar o ruído.



luiz.andrade@tevec.com.br

EXPERIMENTOS COM SÉRIES TEMPORAIS

Experimento 1 – DeNoiser

Set up do experimento

Arquitetura

Duas camadas para encoder e decoder, dimensão do mecanismo de atenção igual a 10 e dimensão das camasdas feedfowrward igual a 100

Otimizador

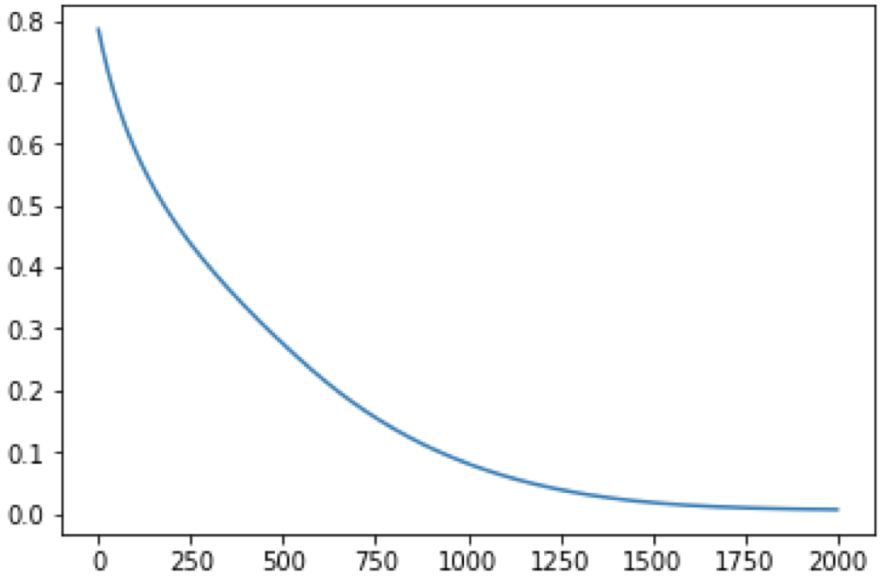
Vanila Stochastic Gradient Descent por 2000 épocas com learning rate de 1e-4

Regularização

Dropout com probabilidade de 0.1 em todas as camadas e layer regularization

Dataset

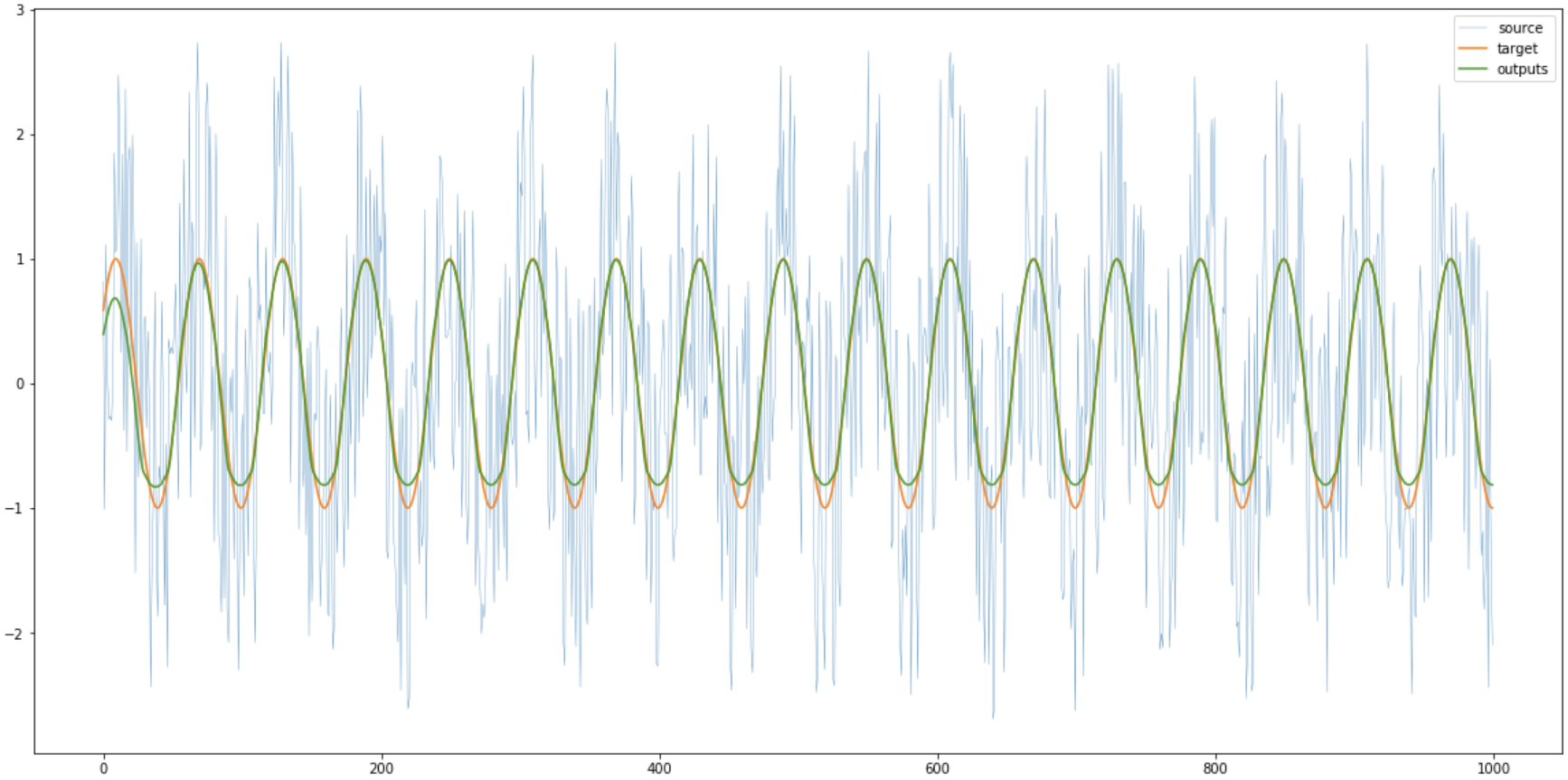
Dataset artificial com 100 observações. Sinal seno com ruído branco entre -1.75 e 1.75



luiz.andrade@tevec.com.br

EXPERIMENTOS COM SÉRIES TEMPORAIS

Experimento 1 – DeNoiser

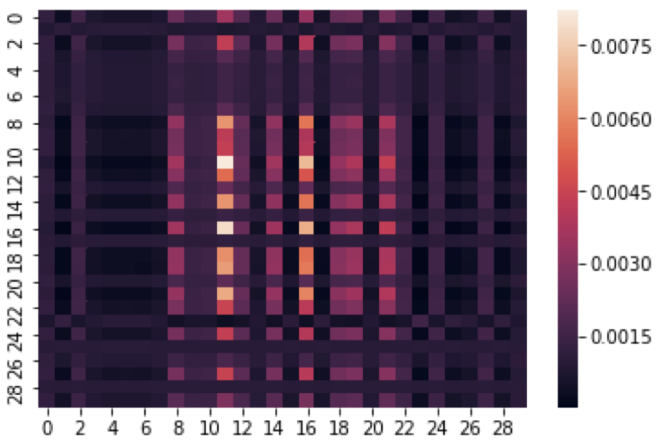


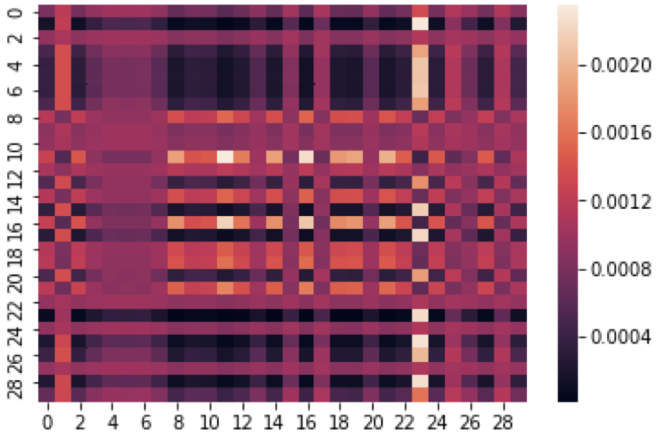
luiz.andrade@tevec.com.br

EXPERIMENTOS COM SÉRIES TEMPORAIS

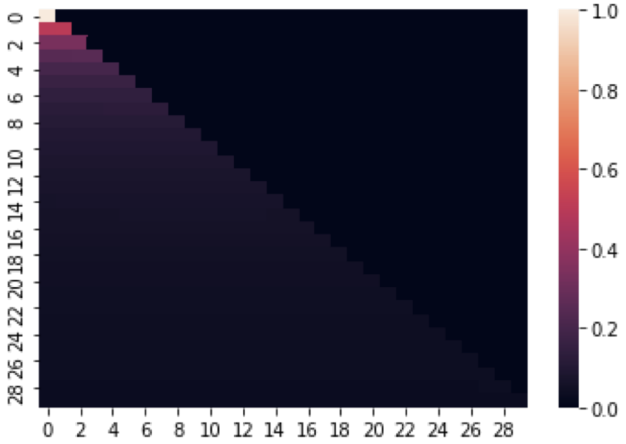
Experimento 1 – DeNoiser

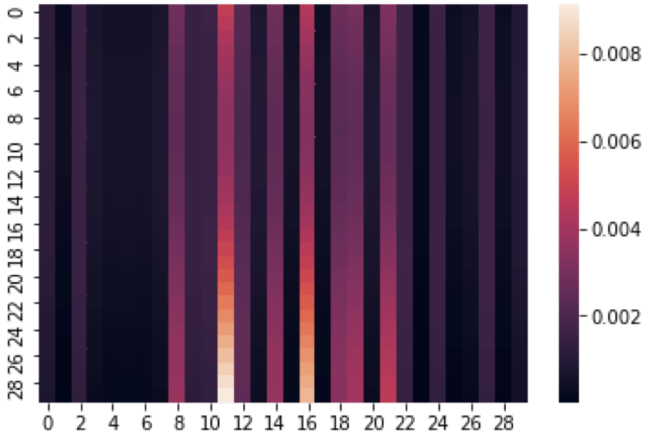
Encoder

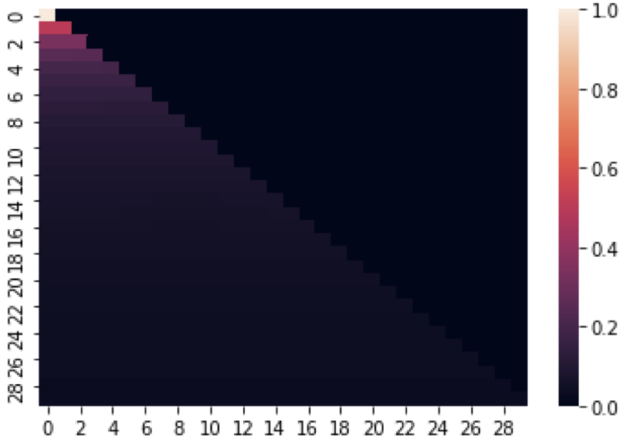


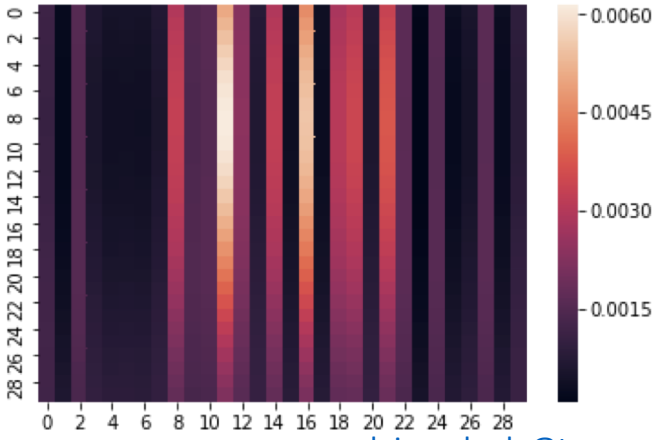


Decoder





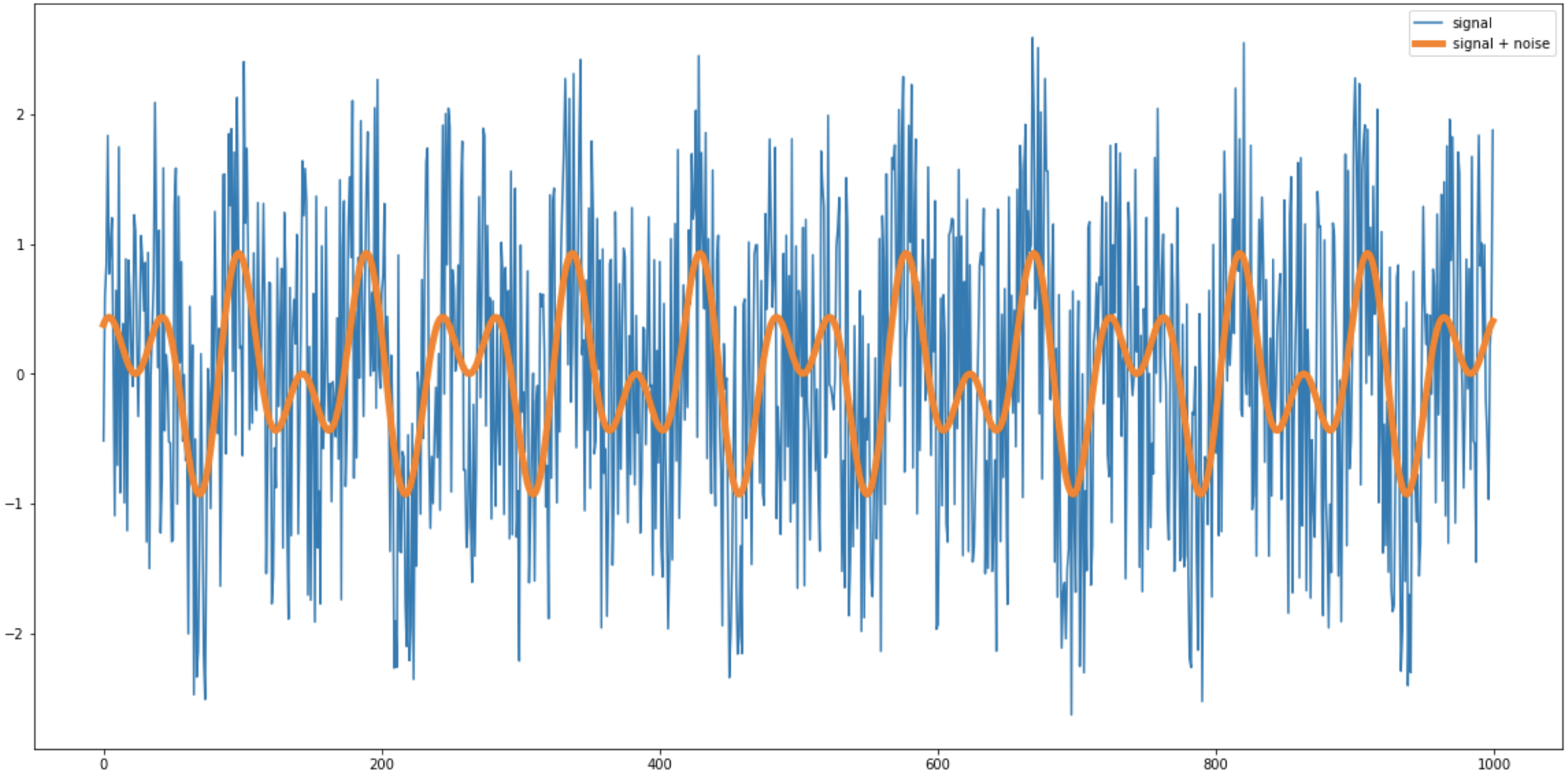




luiz.andrade@tevec.com.br

EXPERIMENTOS COM SÉRIES TEMPORAIS

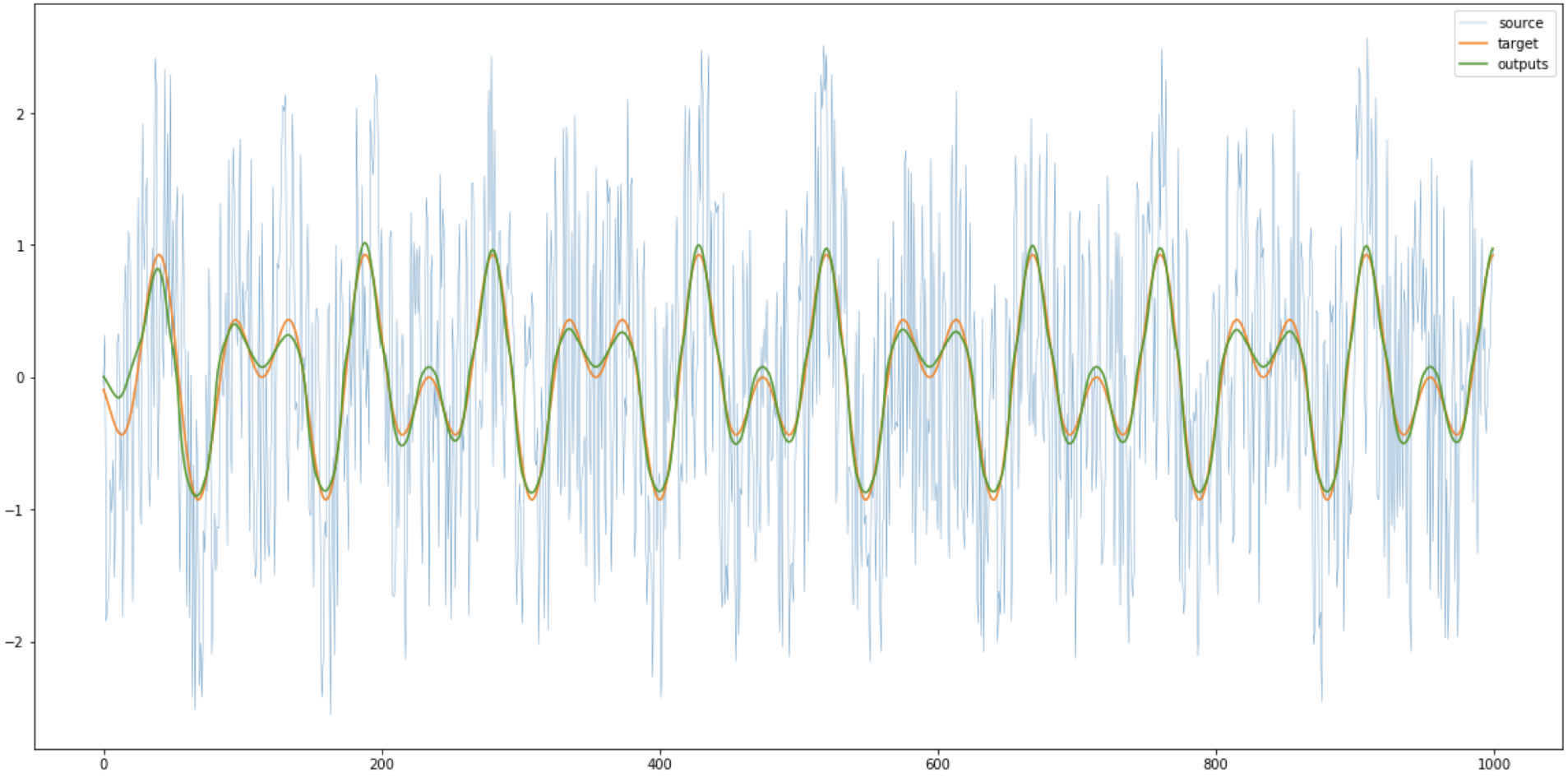
Experimento 1 – DeNoiser



luiz.andrade@tevec.com.br

EXPERIMENTOS COM SÉRIES TEMPORAIS

Experimento 1 – DeNoiser

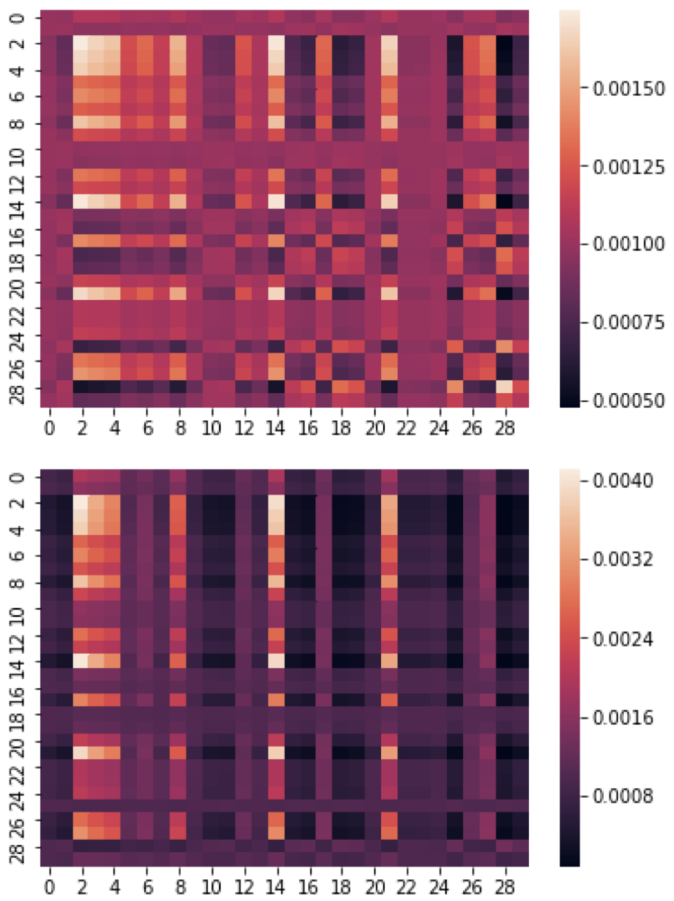


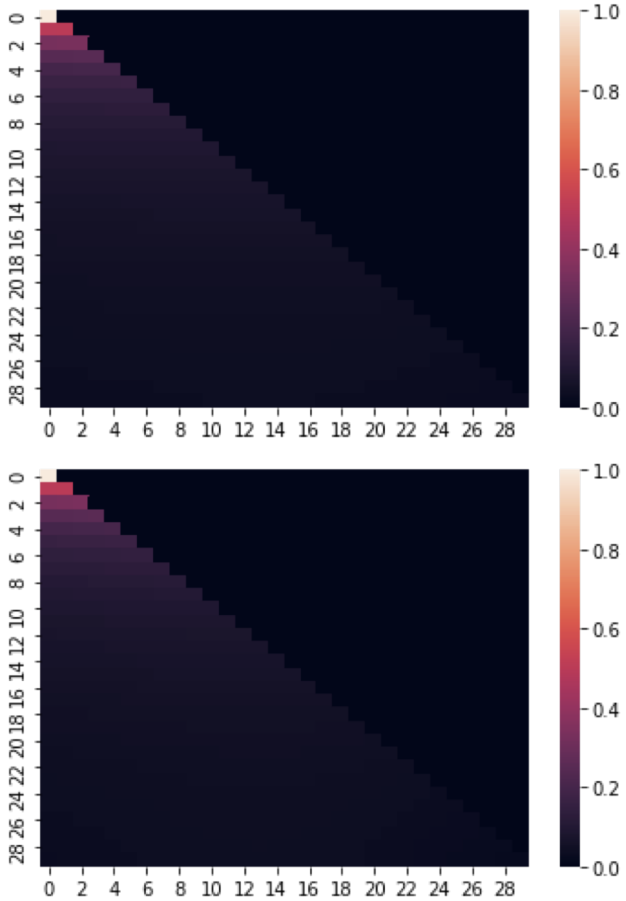
luiz.andrade@tevec.com.br

EXPERIMENTOS COM SÉRIES TEMPORAIS

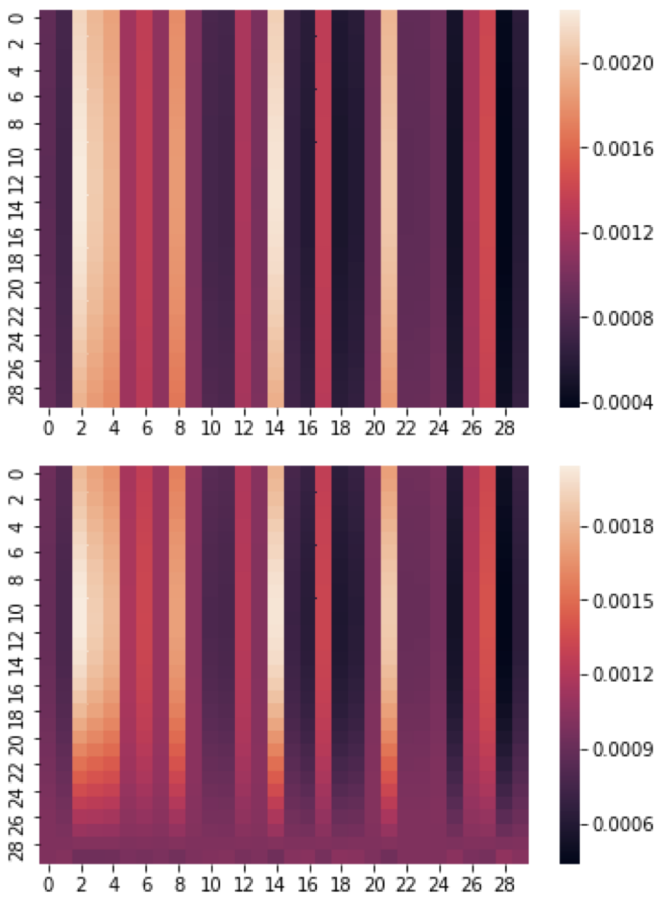
Experimento 1 – DeNoiser

Encoder





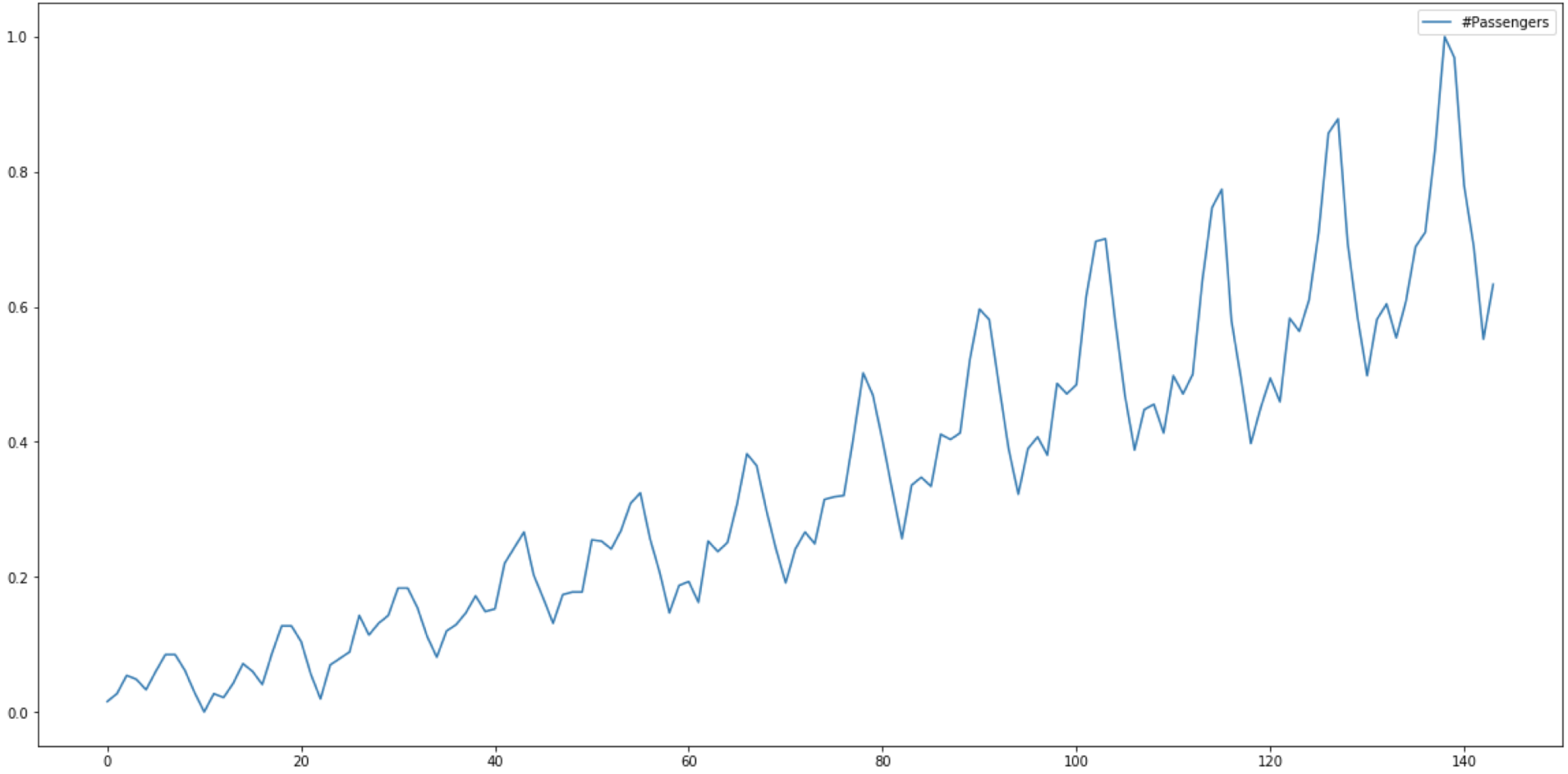
Decoder



luiz.andrade@tevec.com.br

EXPERIMENTOS COM SÉRIES TEMPORAIS

Experimento 2 – Forecast



https://github.com/jbrownlee/Datasets/blob/master/airline-passengers.csv

luiz.andrade@tevec.com.br

EXPERIMENTOS COM SÉRIES TEMPORAIS

Experimento 2 – Forecast

Set up do experimento

Arquitetura

Uma camada para encoder e decoder, dimensão do mecanismo de atenção igual a 40 e dimensão das camasdas feedfowrward igual a 200

Otimizador

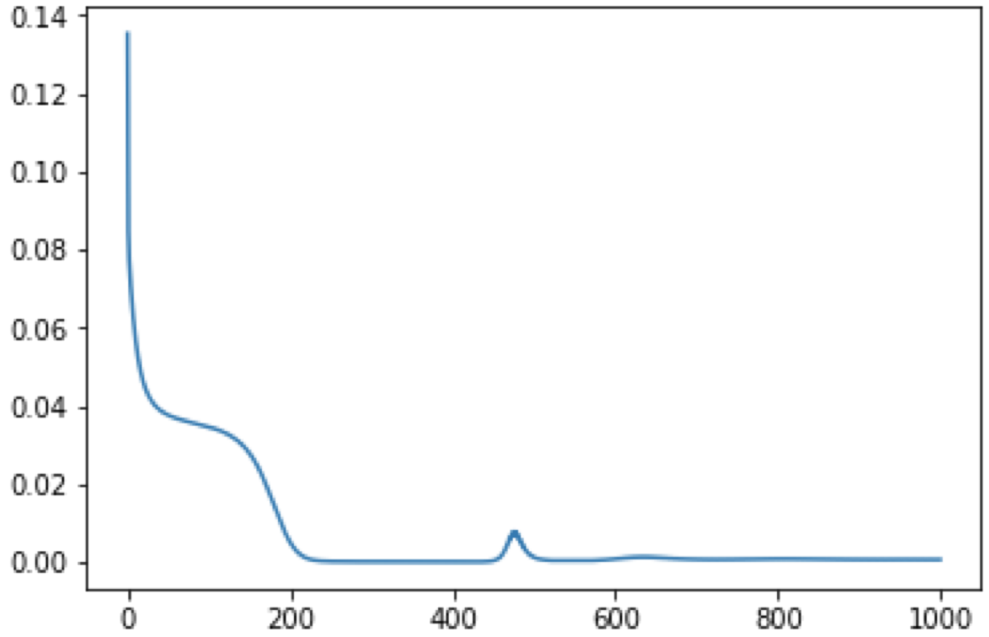
Vanila Stochastic Gradient Descent por 2000 épocas com learning rate de 1e-2

Regularização

Sem dropout e com layer regularization em todas as camadas

Dataset

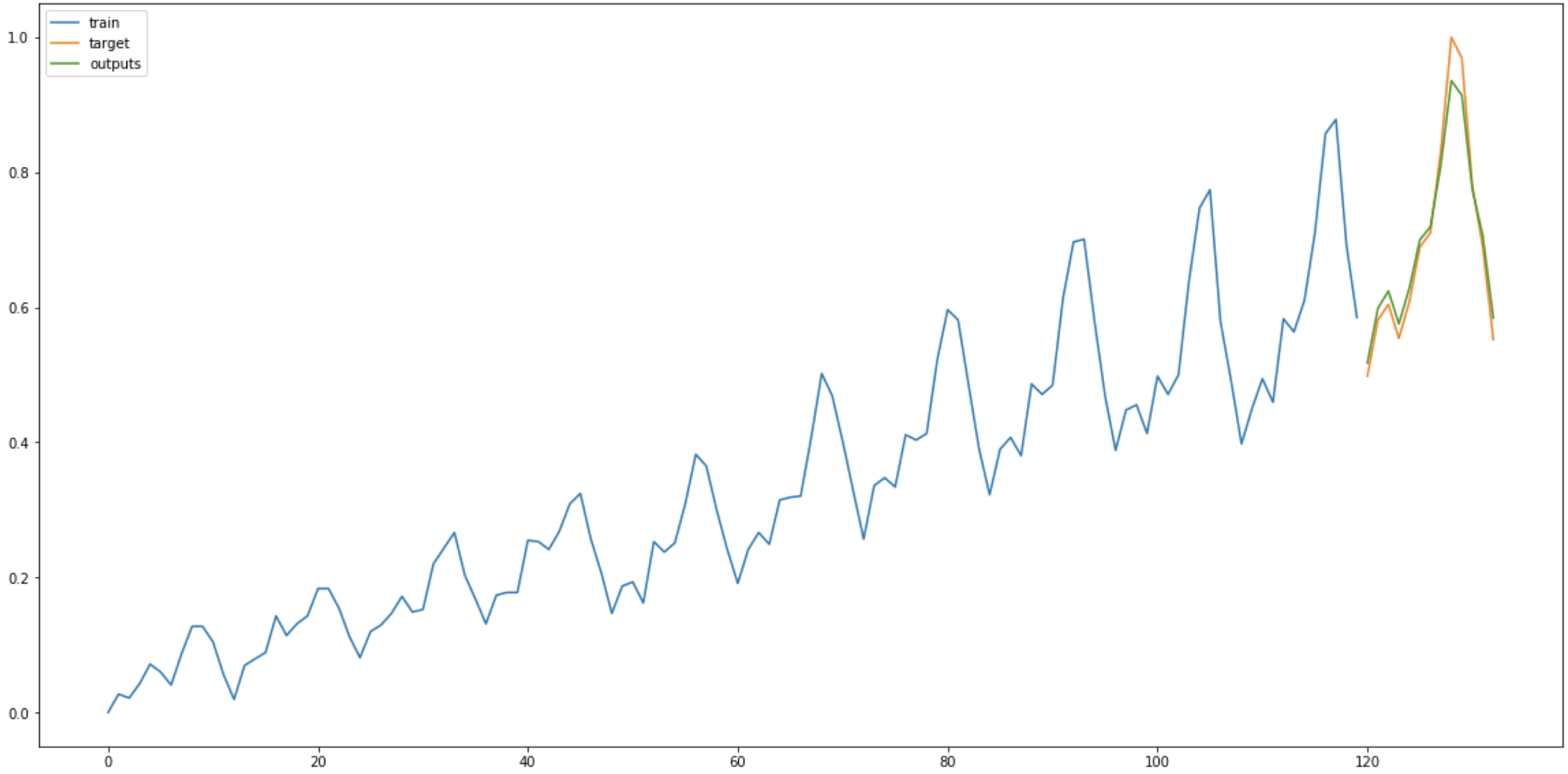
Dataset real, 120 observações de treino e 13 observações de teste



luiz.andrade@tevec.com.br

EXPERIMENTOS COM SÉRIES TEMPORAIS

Experimento 2 – Forecast

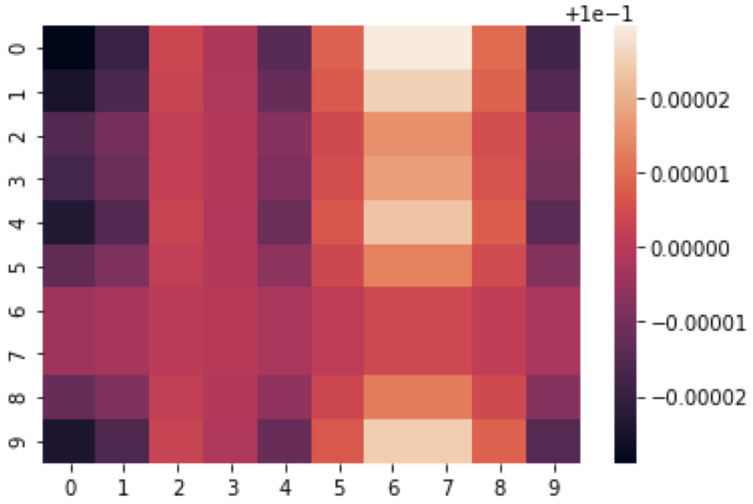


luiz.andrade@tevec.com.br

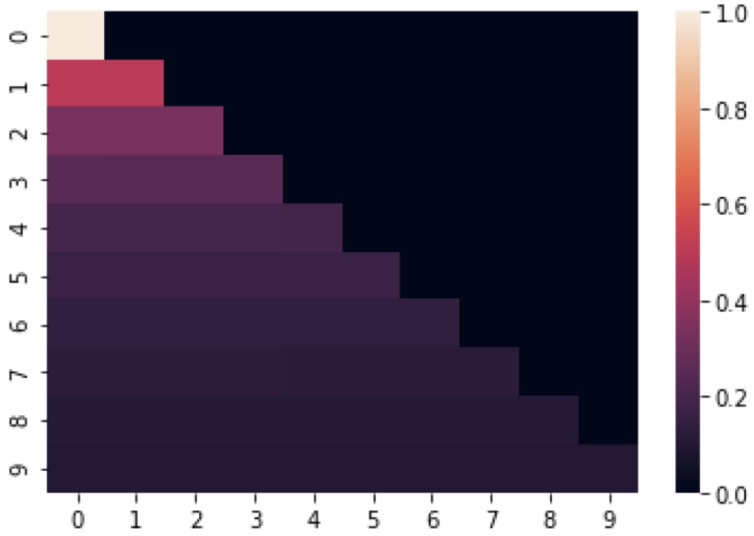
EXPERIMENTOS COM SÉRIES TEMPORAIS

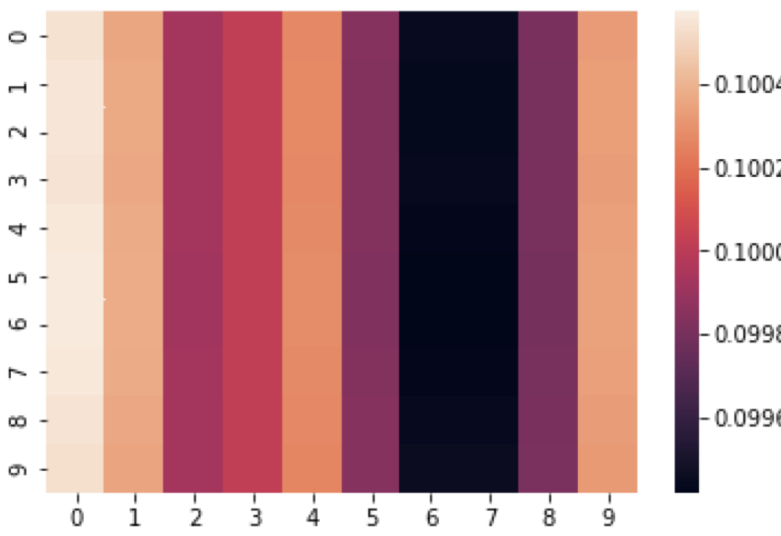
Experimento 2 – Forecast

Encoder



Decoder

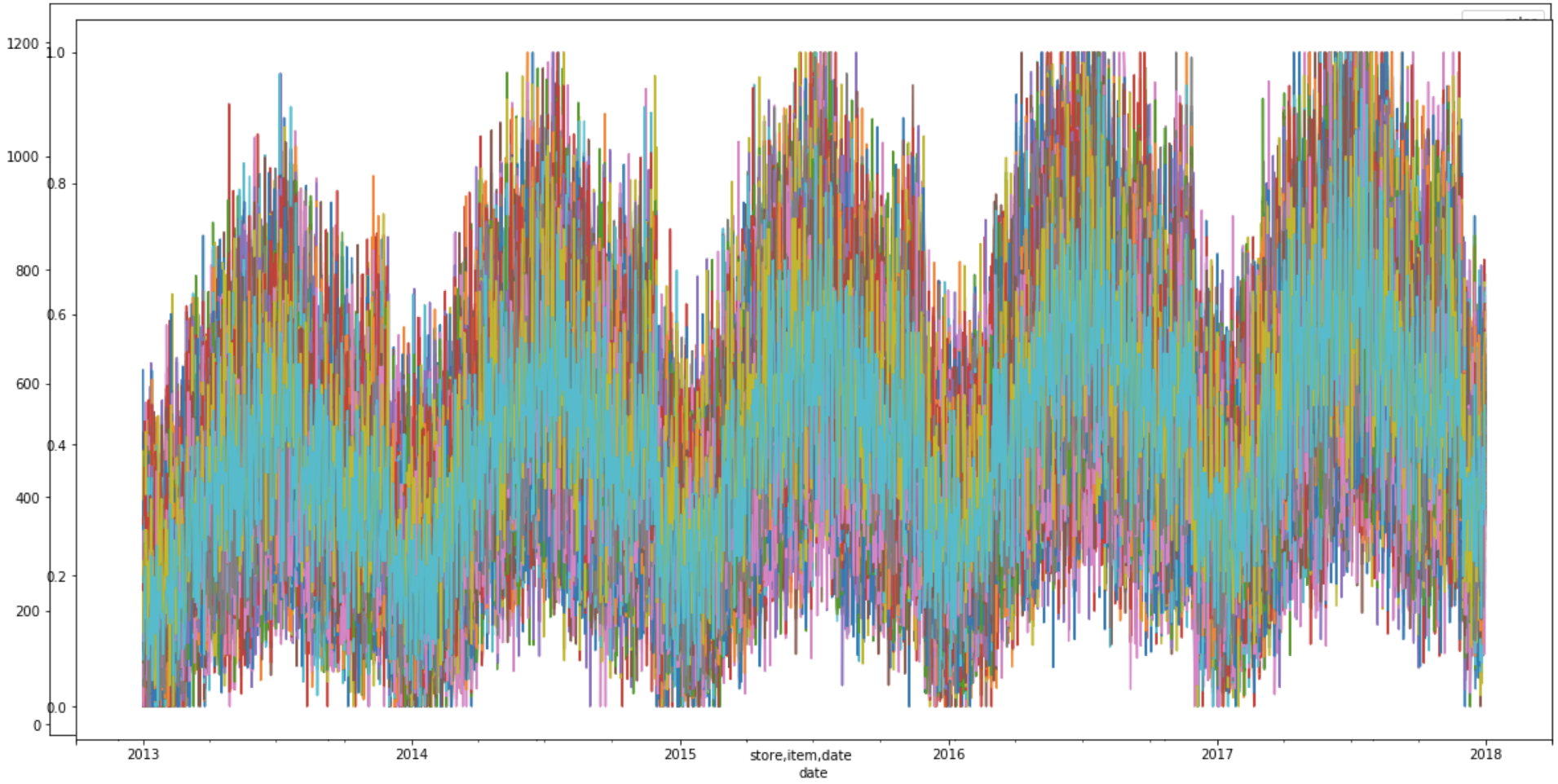




luiz.andrade@tevec.com.br

EXPERIMENTOS COM SÉRIES TEMPORAIS

Experimento 3 – Real Data Forecast



https://www.kaggle.com/c/demand-forecasting-kernels-only/data

luiz.andrade@tevec.com.br

EXPERIMENTOS COM SÉRIES TEMPORAIS

Experimento 3 – Real Data Forecast

Set up do experimento

Arquitetura

Uma camadas para encoder e decoder, dimensão do mecanismo de atenção igual a 500 e dimensão das camasdas feedfowrward igual a 5000

Otimizador

Vanila Stochastic Gradient Descent por 4000 épocas com learning rate de 1e-2

Regularização

Sem dropout e com layer regularization em todas as camadas

Dataset

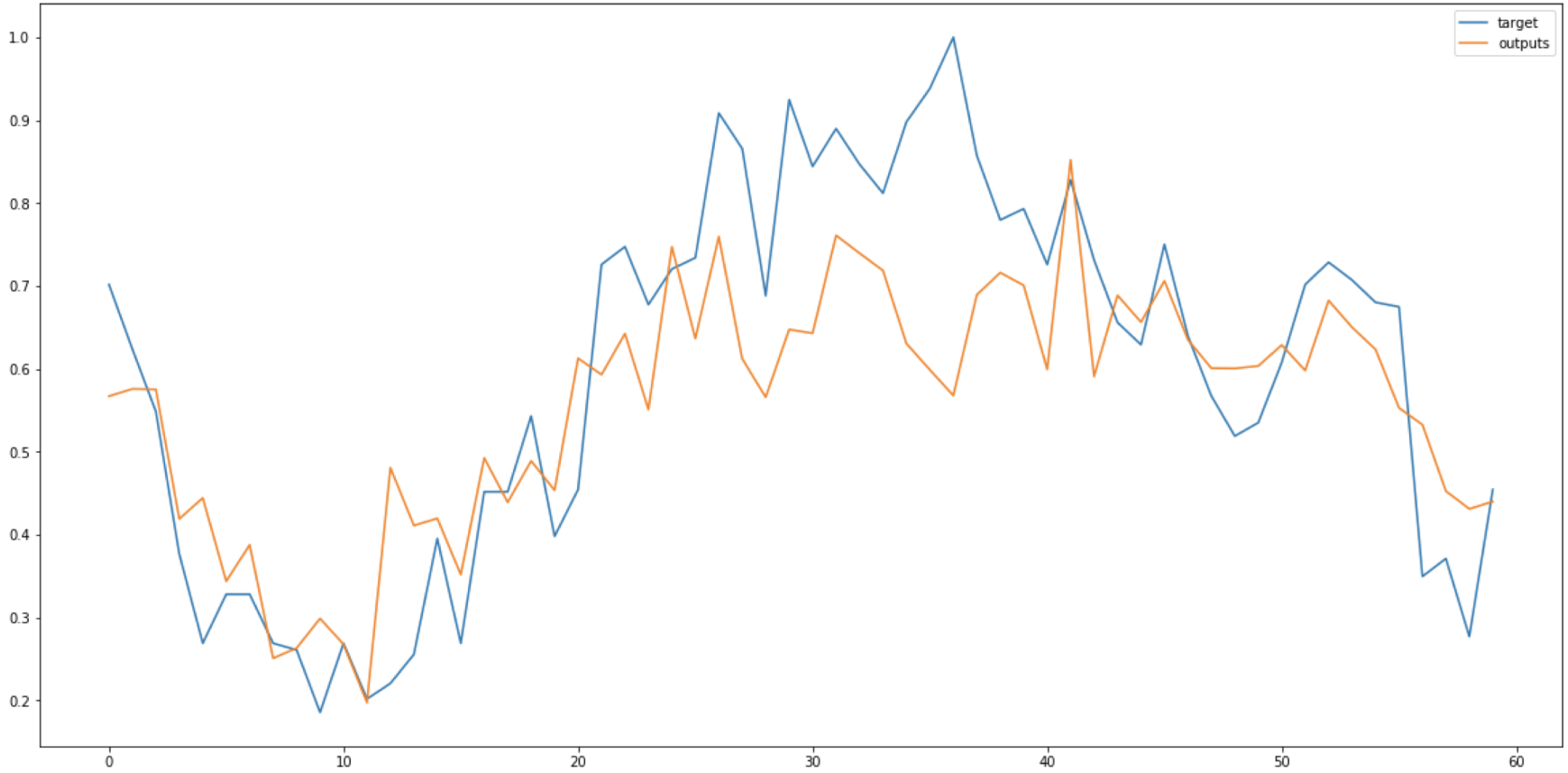
Dataset real, 130500 (50 lojas e 50 produtos) observações

luiz.andrade@tevec.com.br

EXPERIMENTOS COM SÉRIES TEMPORAIS

Experimento 3 – Real Data Forecast

Exemplo de saída

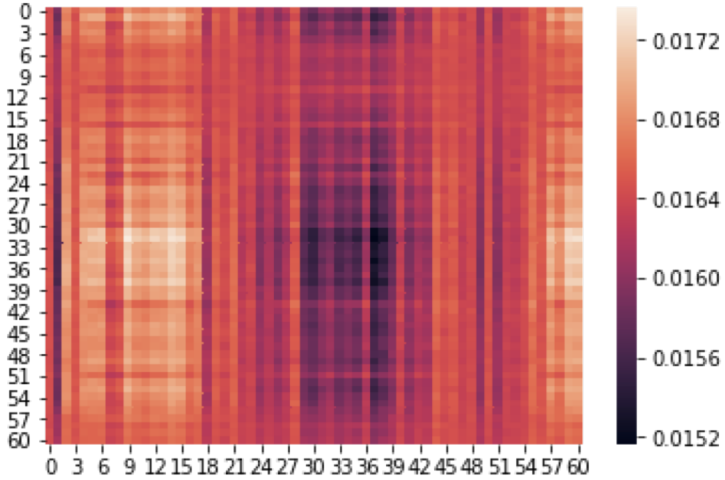


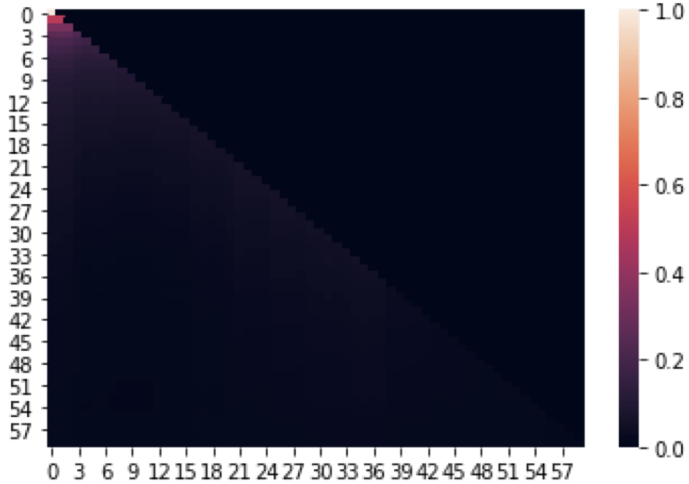
luiz.andrade@tevec.com.br

EXPERIMENTOS COM SÉRIES TEMPORAIS

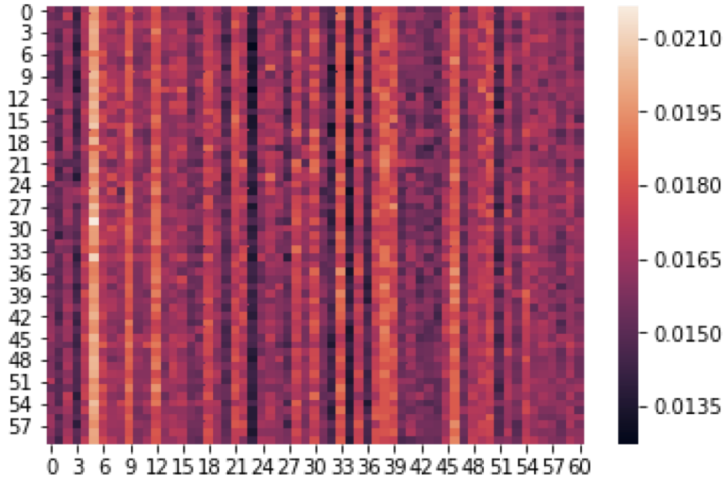
Experimento 3 – Real Data Forecast

Encoder





Decoder

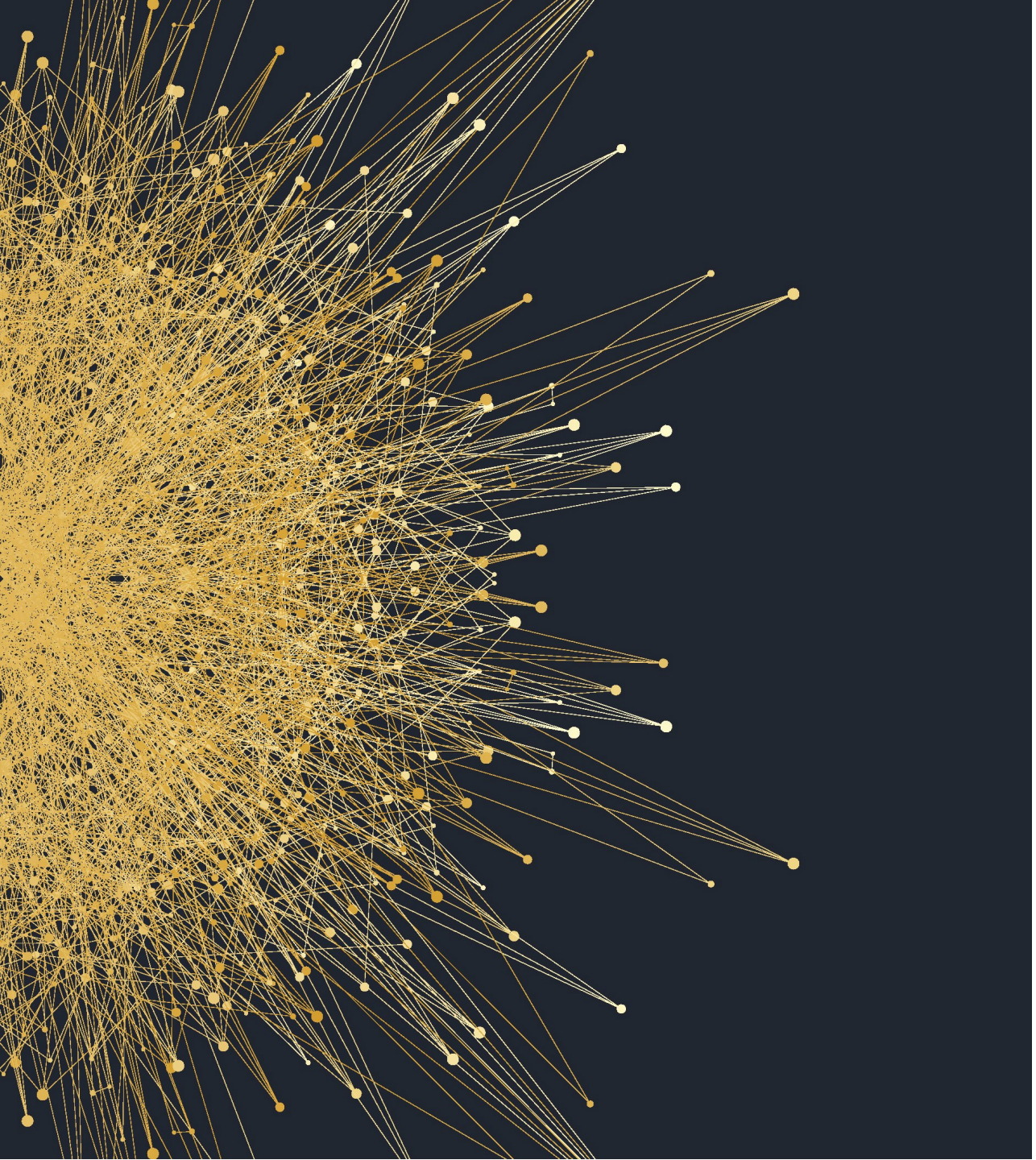


luiz.andrade@tevec.com.br

TAKE AWAY

- O MODELO PODE SER APLICADO A SÉRIES TEMPORAIS BASTANTE CAÓTICAS COM RESULTADOS BONS.
- COMO QUALQUER MODELO DE DEEPLEARNING, ENCONTRAR OS HIPERPARÂMETROS É UM EXERCÍCIO DE PACIÊNCIA.
- PODEMOS TER UMA PRIMEIRA VISÃO “DO QUE A REDE ESTÁ FAZENDO” COM A VISUALIZAÇÃO DOS MAPAS DE ATENÇÃO.
- TRANSFORMER SÃO UM CAMPO DE PESQUISA BASTANTE ATIVO. OS BENCHMARKS CONTINUAM SENDO BATIDOS.
- A ARQUITETURA TRANSFORMER NÃO TEM NADA A VER COM A SÉRIE TRANSFORMERS®

luiz.andrade@tevec.com.br



Obrigado

CONTATO

luiz.Andrade@tevec.com.br

https://www.linkedin.com/in/luiz-andrade-97306216/

VAGAS

FullStack Developer

Data Scientist