MINISTÉRIO DA CIDADANIA

PÁTRIA AMADA BRASIL GOVERNO FEDERAL

Arquitetura Deep Learning Speech-to-Text

PROPOSTA DE DESENVOLVIMENTO

HISTÓRICO

Data	Versão	Descrição	Autor
07/10/2019	1.0	Proposta de desenvolvimento.	Marcello S. Pinheiro

ÍNDICE

INTRODUÇÃO

O presente documento visa mostrar a arquitetura Deep Learning (DL) Speech-to-Text inicial para o projeto Fala Cidadão.

Trata-se de uma arquitetura prevista para funcionar em um Workstation com ou sem a utilização de GPU.

A arquitetura Deep Learning mostrada a seguir, ainda em fase de elaboração e aperfeiçoamento, propõe o desenvolvimento de uma Rede Neural Recursiva denominada Long Short-Term Memory (LSTM) a ser explicada nos capítulos a seguir.

1. AS REDES NEURAIS RECURSIVAS

Segundo DI, BHARDWAJ e WEI em [10], as Redes Neurais Recursivas, do inglês *Recursive Neural Networks* (RNN), estão entre as arquiteturas Deep Learning mais poderosas existentes, pois permitem tarefas como classificação, etiquetagem de dados sequenciais (PoS-Tagger), geração de sequências de texto (prever a próxima palavra) e conversão de uma sequência de palavras em outra, como a tradução de um idioma para outro.

A maioria das arquiteturas de Redes Neurais Artificiais (RNA), como as Redes Neurais Feedforward [6], são projetadas para que todas as características dos dados de entrada sejam mapeadas para aumentar as chances de obtenção das saídas correspondentes, por exemplo, em modelos de previsão de risco de créditos. É através de um sistema elaborado de treinamento dessas RNA que se busca aumentar a acurácia e a precisão, e diminuir o erro na saída das respostas.

Quando a situação requer examinar a ordem ou natureza sequencial de um texto, por exemplo, *o cidadão aguarda o atendente* ou *o atendente aguarda o cidadão*, significa dizer que o sentido do discurso pode mudar completamente, ou seja, a sequência das palavras está diretamente relacionada ao contexto do discurso. Nessas condições, as RNN são muito apropriadas.

Na grande maioria dos idiomas os textos são lidos seguindo a ordem de leitura da esquerda para a direita. As RNN funcionam de maneira semelhante, observando uma palavra no texto de cada vez, onde uma camada intermediária (oculta) dessa arquitetura percorre os dados ao invés de processá-los tudo de uma vez.

Como as RNN podem processar dados em sequência, é possível usar vetores de de palavras ou sinais de áudio de distintos comprimentos a fim de gerarem saídas, também, de distintos comprimentos. Abaixo seguem diferentes arquiteturas, ou topologias, de uma RNN:

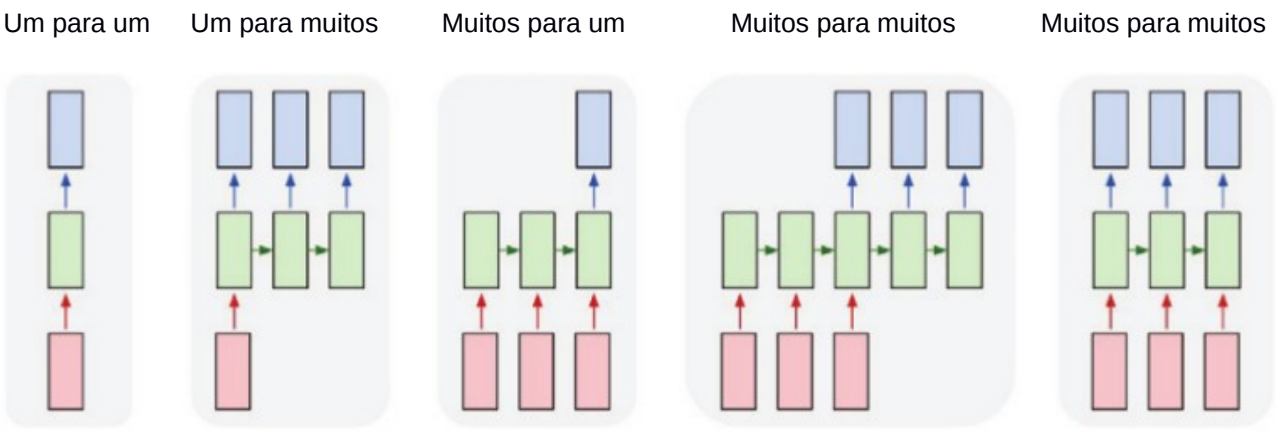


Figura 1 – Tipos de arquiteturas ou topologias de RNN. Imagem retirada de http://karpathy.github.io/2015/05/21/rnn/effectiveness/

2. TOPOLOGIA DEEP LEARNING PROPOSTA PARA O PROJETO FALA CIDADÃO

As RNN, como esclarecido, são utilizadas em situações as quais as sequências de dados da entrada influenciam a resposta da saída. Existem variantes de arquiteturas RNN, explicado em [9], [10], [13], [15] dentre outros, cuja proposta visa resolver problemas de estruturas sequenciais muito longas.

Em ZACCONE e KARIM [15] e em MANASWI [13] mostra que um dos problemas que envolve a utilização das RNN está relacionado ao treinamento delas. RNN possuem uma característica de aprender na medida em que os dados de entrada são imputados. Devido a isso, podem haver desaparecimentos do aprendizado causados por descalibramento no fator denominado gradiente, que ora pode declinar para valores próximos de zero, ora aumentar para valores muito altos, durante o processo contínuo de aprendizagem dessas RNN. A consequência disso é que pode ocorrer uma curta memorização ou uma longa memorização. Em ambos os casos esse é o desafio a ser suplantado.

Nesse sentido, existe uma variação das RNN denominada Long Short-Term Memory (LSTM), pormenorizada em [1], [3], [4], [6], [8], [9], [10], [11], [13] e [15], cujo principal objetivo é o de ajustar o fator de “esquecimento” logrados no processo contínuo de aprendizado nesses tipos de arquiteturas Deep Learning.

A LSTM é um tipo especial de RNN capaz de aprender a dependência entre as sequências de dados de entrada ao longo do tempo. Isso significa dizer que foram desenvolvidas para “lembrarem” de tais informações por períodos curto de tempo [11], [13] e [15]. Em sua topologia, nas LSTM existem camadas menores cujo intuito é dar independência intra camadas evitando, dessa forma, o problema de “esquecimento” por períodos curtos ou longos acima explicado.

Dessa maneira, a LSTM é muito apropriada para ser utilizada em tarefas de Speech-to-Text do projeto Fala Cidadão, pois existem termos nos discursos dos atendimentos que serão, invariavelmente, sempre utilizados e que deverão estar associados (memorizados) a possíveis novos termos tornando-se, dessa maneira, uma solução que possibilita o seu uso nas tarefas de transcrição de áudios para texto, pois a LSTM irá aprender que novos termos se relacionam, também, com os termos mais frequentes permitindo que se ajustem, conforme o input de dados, em períodos adequados de memorização ao longo do tempo.

Por exemplo, termos frequentes como bolsa família, benefício, FGTS, CPF dentre outros serão associados em conjunto com novos termos ou termos menos frequentes. Essa característica permite melhorar gradativamente a acurácia, assim como a precisão, sem perder a memória dos termos menos utilizados ao longo do tempo.

Em outras palavras, a LSTM pode aprender a manter apenas informações relevantes para fazer previsões e esquecer dados não relevantes. A seguir será explicada e ilustrada a topologia LSTM do presente trabalho.

3. TOPOLOGIA LSTM SPEECH-TO-TEXT PARA O PROJETO FALA CIDADÃO

O desenvolvimento da LSTM Speech-to-Text procede dos seguintes passos [3], [6], [10], [13] e [15]:

- Obtenção das características que identificam os componentes de uma onda de áudio, os quais são utilizados para reconhecer o conteúdo linguístico (fala) e excluir os ruídos de fundo inúteis;
    - A fala de uma pessoa qualquer é filtrada pelo formato do trato vocal, pela língua e pelos dentes. O som que está saindo depende desse formato;
    - É necessário determinar esse formato para identificar com precisão o fonema produzido;
    - O formato do trato vocal que se manifesta forma um espectro de potência de curto prazo, ou seja, de ondas curtas;
    - O trabalho das MFCCs (Mel-frequency Cepstral Coefficients) é representar essa forma com precisão.
        - No Wikipedia [16] é explicado que o MFCCs é uma representação do espectro de potência de curto prazo de um som, a qual se baseia na transformação do cosseno linear de um espectro de potência *log* em uma escala de frequência. Os MFCCs são as aplitudes do espectro resultante.
- A fala pode ser representada como dados por meio da conversão da fala em uma matriz com taxa de amostragem;
- A escala mel relaciona a frequência percebida de um tom puro com a frequência real medida;
- Os áudios podem ser convertidos em escalas de frequência a partir de um tom puro para uma frequência mensurada utilizando a fórmula:

$$M(f) = 1125 \ln(1+f/700)$$

Feita a conversão, o estado da célula LSTM atua como uma via de transporte que transfere informações relativas por toda a cadeia de sequência. Funciona como uma "memória" da rede. O estado da célula, em teoria, pode transportar informações relevantes ao longo do processamento da sequência.

As informações das etapas anteriores podem avançar para etapas posteriores, reduzindo os efeitos da memória de curto prazo. À medida que o estado da célula segue sua jornada, as informações são adicionadas ou removidas ao estado da célula através de portas. Os portões são redes neurais diferentes que decidem quais informações são

permitidas no estado da célula. Os portões podem aprender quais informações são relevantes para manter ou esquecer durante o treinamento.

Esses portões contém ativações de função sigmoides. Uma ativação sigmóide é semelhante à ativação da função tangente. Em vez de normalizar valores entre -1 e 1, normaliza valores entre 0 e 1. Isso é útil para atualizar ou esquecer dados, pois qualquer número multiplicado por 0 é 0, fazendo com que os valores desapareçam ou sejam "esquecidos". Qualquer número multiplicado por 1 é o mesmo valor, portanto, esse valor permanece o mesmo ou é "mantido". A rede pode aprender quais dados não são importantes, portanto, podem ser esquecidos ou quais dados são importantes para serem memorizados.

Portanto, são três diferentes portas que regulam o fluxo de informações em uma célula LSTM: o portão para esquecer, o portão de entrada e portão de saída.

O portão de esquecimento decide quais informações devem ser descartadas ou mantidas. As informações do estado oculto anterior e as informações da entrada atual são transmitidas pela função sigmoide. Os valores aparecem entre 0 e 1. Quanto mais próximo de 0 significa esquecer, e quanto mais próximo de 1 significa manter.

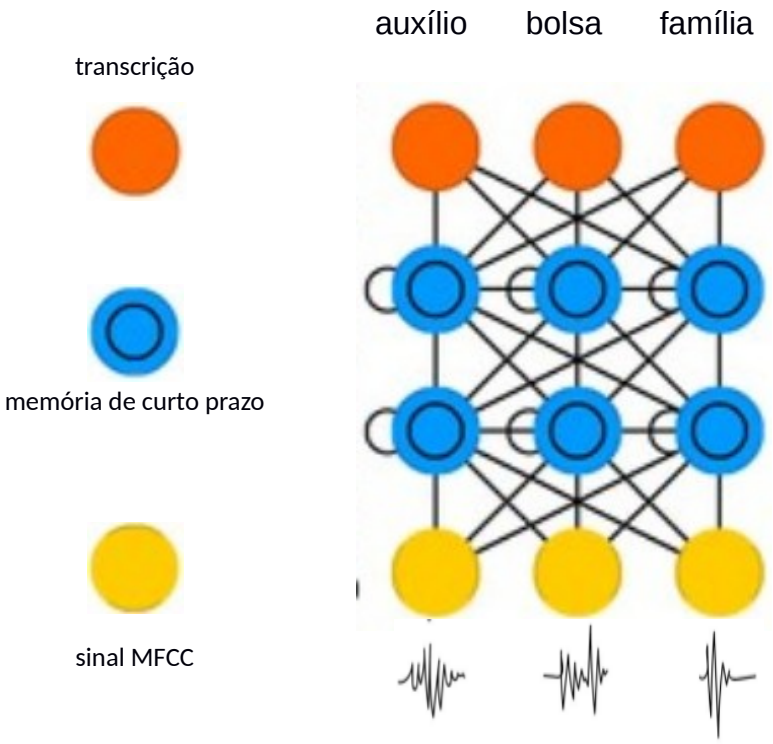
O portão de entrada atualiza o estado da célula. Então, primeiro é passado o estado oculto anterior e a entrada atual para uma função sigmoide, cujo efeito é decidir quais valores serão atualizados, transformando-os entre 0 e 1. O “0” significa não importante e “1” significa importante. Também é passado o estado oculto e a entrada atual para a função tangente visando normalizar os valores entre -1 e 1 para ajudar a regular a rede. Então, multiplica-se a saída tangente pela saída sigmoide. A saída sigmoide decidirá que informação é importante manter da saída tangente.

Dessa maneira, há informações suficientes para calcular o estado da célula. Primeiro, o estado da célula é multiplicado por pontos pelo vetor de esquececimento. Isso tem a possibilidade de descartar valores no estado da célula se multiplicado por valores próximos de 0. Em seguida, é usada a saída do portão de entrada e é feita uma adição cujo objetivo é atualizar o estado da célula para novos valores que a rede neural considera relevantes dando, dessa forma, um novo estado celular.

Por último, o portão de saída decide qual deve ser o próximo estado oculto. Lembrando, o estado oculto contém informações sobre as entradas anteriores. O estado oculto também é usado para previsões. Primeiro, passa o estado oculto anterior e a entrada atual para uma função sigmoide. Em seguida, passa o estado da célula recém-modificada para a função tangente. Multiplica-se a saída tangente pela saída sigmoide para decidir quais informações o estado oculto deve considerar. A saída é o estado oculto. O novo estado da célula e o novo estado oculto são transferidos para a próxima etapa do tempo.

Em síntese, o portão de esquecimento decide o que é relevante para evitar as etapas anteriores. O portão de entrada decide quais informações são relevantes para adicionar a partir da etapa atual. O portão de saída determina qual deve ser o próximo estado oculto.

4. ILUSTRAÇÃO DA TOPOLOGIA LSTM SPEECH-TO-TEXT



O fluxo de controle da LSTM processa os dados transmitidos pelas informações a medida que se propaga. A diferença entre RNN e LSTM são as operações nas células LSTM. Essas operações são usadas para permitir que o LSTM memorize ou esqueça as informações.

Figura 2 - Topologia LSTM

CONCLUSÃO

As RNN são boas para processar dados sequenciais e fazer previsões, mas sofrem com a memória de curto prazo. A LSTM foi desenvolvida como um método para mitigar a memória de curto prazo usando mecanismos chamados portais. Os portais são RNA que regulam o fluxo de informações, as quais fluem através da cadeia de sequência.

A LSTM é usada em aplicações avançadas de Deep Learning, como reconhecimento de voz, síntese de fala, processamento de linguagem natural, análise de sentimentos, análise de séries temporais, etc.

O desenvolvimento e a utilização dessa tecnologia na transcrição dos áudios dos atendimentos para arquivos em formato texto, conforme mostrado, é viável e factível de ser feito. Possibilitará o continuo treinamento e aprimoramento da precisão e da acurácia melhorando, assim, a qualidade dos textos transcritos. O custo inicial de desenvolvimento dessa tecnologia será pago antes do final do segundo ano, se comparado ao custo de uso das soluções de mercado, além de possibilitar a independentização do projeto Fala Cidadão dos serviços de transcrição que são pagos pela utilização.

REFERÊNCIAS

[1] BENGFORT, Benjamin, BILBRO, Rebecca and OJEDA, Tony. Applied Text Analysis with Python. O'Reilly, 2018.

[2] BEYSOLOW II, Taweh. Applied Natural Language Processing with Python. Apress, 2018.

[3] LYTICA. Speech-to-Text Recognition. medium.com consultado em 10/09/2019.

[4] SRINIVASA-DESIKAN, Bhargav. A practical guide to text analysis with Python, Gensim, spaCy, and Keras. Packt Publishing, 2018.

[5] PERKINS, Jocob. Python 3 Text Processing with NLTK 3 Cookbook. Packt Publishing, ed. 2, 2014.

[6] BERNICO, Mike. Deep Learning Quick Reference. Packt Publishing, 2018.

[7] NANDI, Amit. Spark for Python Developers. Packt Publishing, 2015.

[8] SHERIF, Ahmed, RAVINDRA, Amrith. Apache Spark Deep Learning. Packt Publishing, 2018.

[9] RAMSUNDAR, Bharath, ZADEH, Reza Bosagh. TensorFlow for Deep Learning: From linear regression to reinforcement learning. O'Reilly, 2018.

[10] DI, Wei, BHARDWAJ, Anurag and WEI, Jianing. Deep Learning Essentials. Packt Publishing, 2018.

[11] ALBON, Chris. Machine Learning with Python Cookbook. O'Reilly, 2018.

[12] DAS, Sibanjan. CAKMAK, Umit Mert. Hands-On Automated Machine Learning. Packt Publiching, 2018.

[13] MANASWI, Navin Kumar. Deep Learning with Applications Using Python. Apress, 2018.

[14] ANKAN, Ankur. PANDA, Abinash. Hands-On Markov Models with Python. Packt Publishing, 2018.

[15] ZACCONE, Giancarlo, KARIM, Md. Rezaul. Deep Learning with TensorFlow, 2ed. Packt Publishing, 2018.

[16] Mel-frequency cepstrum https://en.wikipedia.org/wiki/Mel-frequency_cepstrum acessado em 14/10/2019.