PRODUÇÃO TEXTUAL UTILIZANDO REDES NEURAIS RECORRENTES

João P A Xavier
joaopedroaxavier@gmail.com

Mateus C L Castro
samuraiexx@gmail.com

Natália F G Souza
natalia_godot@hotmail.com

5 de dezembro, 2016

Resumo

Serão apresentados os conceitos de Aprendizado de Máquina, Aprendizado Profundo, Redes Neurais Recorrentes e *Long Short Term Memory*, bem como a utilidade destes no âmbito de problemas de classificação/regressão em geral. Posteriormente, será investigada a capacidade das LSTMs de gerar sentenças de texto em escala de caractere a caractere, avaliando quantitativamente e qualitativamente o resultado de modo a encontrar a estrutura que é mais capacitada a gerar um texto suficientemente próximo da escrita real em Língua Portuguesa.

Abstract

The concepts of Machine Learning, Deep Learning, Recurrent Neural Networks and Long Short Term Memory will be presented, as well as the utility of them in general regression/classification problems. Then, the LSTMs aptitude to generate text sentences in character-to-character scale will be investigated by quantitative and qualitative analysis to find the structure in which there is the better fitting to real written Portuguese Language texts.

1 Introdução

1.1 Contexto

Inteligência artificial é um ramo da ciência que visa a compreender e construir entidades inteligentes, ou seja, capazes de tomar decisões por si próprias baseadas em uma série de parâmetros. É uma área que vem crescendo recente e de grande aplicabilidade em diversos campos como robótica, mineração de dados, jogos, reconhecimento de voz, diagnóstico de doenças, entre outras.[3]

Machine Learning é um método de inteligência artificial muito utilizado na atualidade para selecionamento e análise de dados por máquinas. Seu objetivo é permitir que o computador adote as próprias decisões baseado em resultados anteriores e em um conjunto de dados, aprimorando o processo a medida que é exposto a novos grupos de entrada.

A técnica baseia-se na utilização de diversos grupos de teste a partir dos quais o computador otimiza seu algoritmo de análise. A máquina então é apresentada a um novo grupo desconhecido e, com base no algoritmo otimizado, classifica a nova entrada. As aplicações são diversas, desde reconhecimento facial ou de digitais à identificação de padrões de voz, de imagens e mecanismos de busca.

1.2 Motivação

Apesar de sua grande aplicabilidade, o *machine learning* apresenta limitações. É necessário um profissional de elevado nível técnico para sua utilização, uma vez que as variáveis de maior relevância ao problemas precisam ser definidas pelo usuário e os dados de entrada devem ser pré-processados antes de serem enviados ao algorítmo. O *deep learning* permite que os dados sejam fornecidos em sua forma bruta e que as variáveis mais relevantes ao problema sejam estabelecidas pelo próprio algorítimo, dessa forma dispensando que o usuário tenha conhecimento em estatística para tratamento de dados.

Logo houve motivação para o desenvolvimento de uma nova arquitetura mais complexa, o *deep learning*. Essa técnica dispensa domínio técnico e trabalho necessários no pré-processamento dos dados em métodos tradicionais de Machine Learning.

1.3 Aprendizado Profundo

O Aprendizado Profundo, ou *Deep Learning*, consiste em um conjunto de algoritmos em *Machine Learning* inspirado no cérebro humano e capaz de lidar com os dados na forma bruta, sem necessariamente precisar de um pré-processamento. Por exemplo: Para reconhecimento de face, é possível aplicar o *Deep Learning* utilizando como dados para treino um grande conjunto de fotografias de rostos (Há grandes trabalhos para construção de datasets grandes, como por exemplo o conjunto de mais de 13.000 fotos de rostos[6] produzido em 2007 pela Universidade de Massachusetts). Para lidar com esse tipo de problema, a estrutura do *Deep Learning* normalmente é formada por várias camadas de "neurônios"nas quais os dados são processados com transformações lineares e não lineares. Essas transformações dentro dos "neurônios"podem usar vários tipos de funções, pegando como parâmetro os dados que vieram da camada anterior. O tratamento de imagem não seria possível com o método convencional do *machine learning*, uma vez que a estrutura mais simples não permite o uso de diversos níveis de abstração, bem como dificulta o tratamento de

um grande volume de dados.

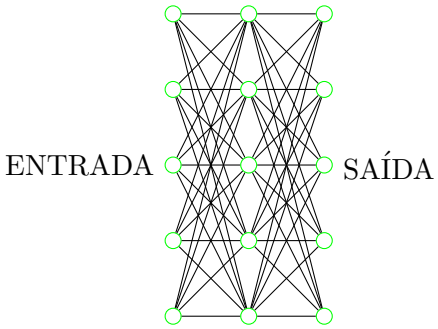

Figura 1: Exemplo visual de uma rede neural com três camadas.

Um parâmetro bastante importante para que a rede seja otimizada é a *função custo*. Esse parâmetro indica o quão próximo dos exemplos de treino (i.e. os conjuntos de dados usados no treino) a saída da rede está. Durante o treinamento da rede, deseja-se minimizar o valor do custo de forma que ela possa gerar saídas que sejam condizentes com os valores reais. Contudo, espera-se que a rede não fique limitada aos casos dados no treinamento, que é o que ocorre quando o custo fica muito baixo, causando o *overfitting* da rede.

A função custo pode ser computada por meio de diversos modelos. Entre eles, está a Entropia Cruzada (em inglês *Cross Entropy*), fortemente relacionada à função sigmóide (ou função logística). No aprendizado supervisionado, a *Crosss Entropy* e a Sigmóide caracterizam o que se chama de "Regressão Logística"ou "Regressão Binária Logística"(mais comumente em inglês, por *Logistic Regression*):

$$\sigma(\theta^T x) = \mathbb{P}(y = 0) = \frac{1}{1 + e^{-\theta^T x}} \qquad (1)$$

$$\mathbb{P}(y = 1) = \frac{e^{-\theta^T x}}{1 + e^{-\theta^T x}} \qquad (2)$$

Sendo θ^T o vetor utilizado como "peso", ajustado no algoritmo de propagação. A saída na distribuição acaba sendo binária nesse caso (pois as possíveis saídas são "0"ou "1").

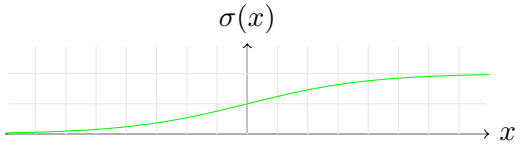

Figura 2: Função Sigmóide ou Logística

Quando a saída é um conjunto de estados (não binário, como é no caso da Regressão Logística), é conveniente que exista uma adaptação da Regressão Logística para o caso multivariável. Para tal, existe a Regressão Multinomial Logística (conhecido em inglês por *Multinomial Logistic Regression*, também conhecida como Regressão *Softmax*. Nesse tipo, espera-se uma "distribuição de probabilidades"para os N estados como a apresentada a seguir.

$$\mathbb{P}(y = k) = \frac{e^{-\theta_k^T x}}{\sum_{i=1}^{N} e^{-\theta_i^T x}} \qquad (3)$$

Para um dado vetor x. Sendo $\theta_1^T, \theta_2^T, ..., \theta_N^T$ os N vetores utilizados como peso. Note que para o caso N=2 teremos, como esperado, a Regressão Logística. A Regressão *Softmax* é utilizada nos casos de produção de texto, por exemplo. Nesse caso, os vários estados correspondem às diferentes letras conhecidas do "dicionário"da rede. O custo no caso da Regressão Binária Logística é calculado da seguinte forma:

$$L = -\frac{1}{n} \sum_{x} y ln(\bar{y}) + (1 - y) ln(1 - \bar{y}) \qquad (4)$$

Sendo $\bar{y} = \sigma(\theta^T x)$. Dada uma função custo, os neurônios de cada camada são ajustados por meio de algoritmos de otimização. Alguns exemplos desses algoritmos são o Adam[8], o SGD[1] e o Adadelta[10]. Com a execução dos algoritmos, espera-se que a Rede seja capaz de classificar novos casos diferentes do que foi dado durante a fase de treino.

Embora as redes neurais tradicionais resolvam uma série de problemas, elas são incapazes de "lembrar"do que foi gerado anteriormente e utilizar essa informação para ajustar a saída da rede. Por exemplo: Em um problema de produção de texto, é de interesse que o texto gerado anteriormente influencie de alguma forma na próxima saída da rede. Para tal, a rede neural pode utilizar a entrada comum e a última saída gerada por ela, formando uma entrada um pouco mais sofisticada que o caso comum.

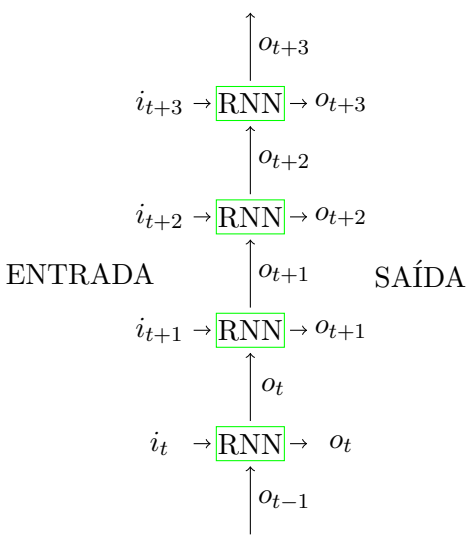

Figura 3: Exemplo visual de uma Rede Neural Recorrente, na qual a saída das iterações é aproveitada no ajuste da rede

No estudo e aplicação das Redes Neurais Recorrentes, é muito comum o uso do termo *Long Short Term Memory* (LSTM)[5], que é a essência desse tipo de rede. As LSTMs são um tipo especial de RNN que introduz de maneira mais eficaz a característica de "memória"da rede, Uma vez que a estrutura tradicional da RNN é incapaz, na prática, de lembrar os dados gerados a longo prazo sem ter grande custo computacional (seja em tempo ou em custo de memória). Em um problema de produção de texto, a recordação da rede influencia diretamente na relação entre as palavras e sentenças e, portanto, afeta a coerência do que é escrito.

1.4 Objetivo

Uma possível aplicação das técnicas de inteligência artificial, em particular das redes neurais recorrentes, é a síntese automática de textos. Com uso das ferramentas de RNN e *Deep Learning*, e em conjunto com um *dataset* de romances de Machado de Assis - usados para treinamento do modelo e avaliação do erro - o objetivo desse trabalho é criar um mecanismo automático de produção de textos literários por uma máquina, sem intervenção direta do ser humano.

1.5 Trabalhos Relacionados

O emprego das Redes Neurais Recorrentes e das LSTMs já é uma realidade em diversas aplicações.

Um exemplo de aplicação é o problema de classificação de imagem por legendas[7], realizado em uma pesquisa de Stanford em 2015. Nela, as "palavras-chave"da descrição são geradas pelo reconhecimento da imagem e, posteriormente, a legenda sai com o auxílio de uma RNN, que possibilita uma construção mais clara e objetiva da imagem de acordo com as "palavras-

chave"geradas.

Outro exemplo é a utilização de LSTMs para improvisação musical (do gênero *Blues*), bem como o estudo de uma estrutura temporal desse tipo de improvisação[2]. As memórias de longo prazo das LSTMs criam uma estrutura global à saída, prezando pela "harmonia"do som gerado.

A utilidade da memória a longo prazo possibilitada pelas LSTMs é muito bem aproveitada em problemas que envolvem manuseio de sequências. Essa utlidade já foi aproveitada na tradução automática do Inglês para o Francês[9]. No caso, duas LSTMs são utilizadas. Em uma, realiza-se a leitura do texto em inglês, enquanto na outra usa a estrutura gerada pela primeira rede para produzir o texto correspondente em francês.

A abordagem quanto à sintese automática de texto não é inédita, e já foi aplicada, por exemplo, na produção de esqueletos de páginas da Wikipedia[4]. Nesse caso o nível de abstração é por palavras, diferente da escala letra-a-letra feita aqui neste trabalho. Contudo, a abordagem do problema é, em geral, bastante similar. Além desta, outras aplicações são realizadas, como a transformação de texto digitado para a forma "manuscrita".

2 Estudo de Caso

A geração de texto, dentre outras aplicações do *deep learning*, pode ser executada de diversas formas, variando: a forma como os dados são fornecidos para a rede; os próprios dados de entrada; a arquitetura da rede neural; o método utilizado para calcular o erro; o método de treinamento da rede; a forma como as saídas da rede são utilizadas para gerar o texto. Neste artigo será seguido um modelo, o qual será descrito, variando-se a arquitetura da rede a fim de observar qual tem melhor desempenho.

2.1 Modelo utilizado

No caso escolhido para estudo de geração de texto, os caracteres do texto alimentam a rede, de tal forma que um determinado número de caracteres até um ponto do texto é a entrada para a rede neural, sendo cada caractere anterior o resultado em um tempo anterior, conforme a figura 4.

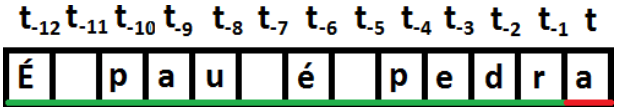

Figura 4: Exemplo visual do mecanismo de geração de texto.

Neste exemplo cada um dos caracteres em verde serve como entrada da rede neural e o caractere 'a' é o gerado na saída, assim como no próximo passo, ocorrerá conforme a figura 5.

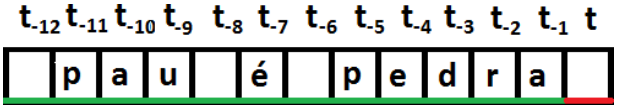

Figura 5: Exemplo visual do mecanismo de geração de texto, uma iteração após a figura 4.

Assim, o caractere gerado anteriormente, 'a', se torna mais um parâmetro de entrada e o caractere espaço é a nova saída da rede, continuando este processo um certo número de vezes pode-se criar frases, parágrafos e até mesmo textos com a rede neural.

A fim de colocar um caractere como entrada ou saída na rede há a necessidade de se fazer a conversão de caractere para algum formato introduzível na rede, nas entradas, ou retornar do

formato usado na rede para o de caractere, nas saídas. No caso do modelo utilizado cria-se um vetor de tamanho equivalente ao número total de caracteres, para cada caractere. Dessa forma cada posição do vetor corresponde a probabilidade do caractere utilizado seja o caractere correspondete àquela posição, nas saídas da rede. Já na entrada os caracteres são definidos, desta forma os vetores se tornam nulos em todas as posições exceto na correspondente ao caractere definido e a partir destas a rede gera a saída com as probabilidades correspondentes, conforme a figura 6.

	p	a	u		é		p	e	d	r	a
É	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	0	0	1	0	1	0	0	0	0	0
p	0	1	0	0	0	0	0	1	0	0	0	0
a	0	0	1	0	0	0	0	0	0	0	0	1
u	0	0	0	1	0	0	0	0	0	0	0	0
é	0	0	0	0	0	1	0	0	0	0	0	0
e	0	0	0	0	0	0	0	0	1	0	0	0
d	0	0	0	0	0	0	0	0	0	1	0	0
r	0	0	0	0	0	0	0	0	0	0	1	0

Figura 6: Exemplo de matriz de probabilidades para caracteres definidos.

Com a finalidade de treinar uma rede, cria-se então uma matriz de 3 dimensões a qual contém uma dimensão correspondente às sequencias, isto é um espaço para cada sentença, e outras duas dimensões, tal como no exemplo acima, correspondentes aos caracteres de cada sentença, sendo esta a matriz 'X' de entrada. Outra matriz de duas dimensões, 'Y', também é criada, no qual uma dimensão, tal qual na primeira matriz, corresponde à sentença da qual será gerado o próximo caractere e outra correspondente ao caractere que se espera ser gerado pela rede, conforme a figura 7:

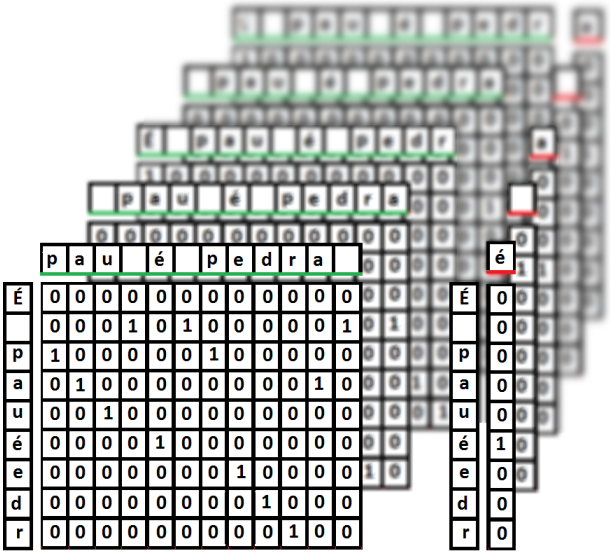

Figura 7: Exemplo de matriz de entrada e matriz de saída da rede neural.

2.2 Experimentos

Com isto a rede é treinada, utilizando um método de optimização de rede e de calculo de erro, desta forma, a rede tenta ajustar os próprios parâmetros de tal forma que as a coluna de probabilidades de saída da rede para uma sequência de 'X' seja tão próxima quanto possível da coluna correspondente em 'Y' para a mesma sequência. A fim de se fazer um acompanhamento da rede, pega-se o erro após cada vez que o algoritmo é treinado com parte do texto. O erro deste treino é computado e passa-se a rede em uma parte do texto a qual não é utili-

zada para treino, computando também esse erro do conjunto de teste, de tal forma que no final do treino um gráfico é gerado por estes dois conjuntos de pontos. A partir destes pode-se fazer uma análise quantitativa e depreender algumas características da arquitetura da rede utilizada, principalmente se houve *overfit* ou *underfit*. O primeiro ocorre quando a rede se acostuma muito com o grupo de treino, assim o erro no grupo de teste será muito maior, enquanto o segundo ocorre quando a rede se acostuma pouco com o grupo de treino, assim o erro do grupo de teste e do grupo de treino serão muito próximos.

A partir da rede treinada, a fim de ter um texto gerado pela rede neural, utiliza-se uma semente, ou seja, um pequeno trecho que será utilizado como entrada no início da geração do texto. Um vetor de probabilidades é então gerado pela rede. Este vetor é modificado de tal forma que cada um dos seus argumentos é elevado a 1/T, onde T representa um parâmetro denominado temperatura. Dessa forma as probabilidades se aproximam quando T é maior que 1, diminuindo a certeza sobre escolher um certo caractere, e se distanciam quando T é menor que 1, aumentando a confiança no caractere cuja probabilidade é maior. Após isto, o vetor é normalizado novamente e o próximo caractere é 'escolhido' ou 'sorteado' tendo como base as probabilidades de cada caractere. Gerando textos desta forma a partir das redes treinadas pode-se fazer uma análise qualitativa destas.

Para este trabalho, foram criadas nove redes neurais. Tomando-se uma destas como base, variou-se as características a fim de identificar a rede mais eficiente na geração de texto. Para melhorar o entendimento, serão explicitadas e explicadas as características, além de como classificar o resultado como mais ou menos eficiente.

Nas imagens de exemplo dadas, cada caractere era previsto a partir dos 12 caracteres anteriores, à este tamanho, dá-se o nome de tamanho de janela. Em vez de treinar a rede com todos os caracteres seguidos pode-se pular um certo número de caracteres até o próximo a servir como saída, a este número se atribui o nome "passo", desta forma pode-se ter uma diversidade maior nos textos usando a mesma quantidade de memória. O erro na previsão de uma sequencia de caracteres é estimado utilizando o algoritmo *Categorical Cross Entropy*, comprando a probabilidade de uma sequência de saídas da rede com as esperadas, em seguida os pesos da rede são atualizados utilizando o algoritmo *Adam* [8], o número de entradas desta sequência é chamado de *batch size*.

Em relação à própria arquitetura da rede, todas seguem o modelo de uma camada LSTM com o número de nerônios definido em cada experimento, seguida de um *dropout* - nesta parte, uma porcentagem definida das saídas da camada anterior são aleatoriamente zeradas, um mecanismo simples para evitar overfitting - e uma camada densa com função de ativação *softmax*.

Os experimentos são todos realizados usando as obras Dom Casmurro, Memórias Póstumas de Brás Cubas e Quincas Borba de Machado de Assis (domínio público), de tal forma que 80% destes textos são utilizados para treinar a rede e os 20% restantes para calcular o erro de teste, de um grupo fora o de treinamento.

Dentre as nove redes criadas, a terceira serviu de base para as demais, isto é, todas as demais são iguais à primeira a menos de uma característica. Elas foram treinadas por trinta e nove épocas, cada época representa um looping do algoritmo de treinamento por todos os três textos. Segue a tabela, figura 8, com as características de cada uma, sendo a característica modificada grafada em vermelho.

Experimento	1	2	3	4	5
Tamanho de janela	100	100	100	60	120
Neurônios	256	256	256	256	256
Dropout	0.2	0.7	0.5	0.5	0.5
Batch size	128	128	128	128	128

Experimento	6	7	8	9
Tamanho de janela	100	100	100	100
Neurônios	256	256	512	128
Dropout	0.5	0.5	0.5	0.5
Batch size	256	64	128	128

Figura 8: Tabela contendo as características de cada experimento.

3 Resultados e Discussões

Como o experimento 3 foi tomado como base, este será o primeiro a ser analisado, com o fim de poder comparar os demais a este. O texto gerado para a análise qualitativa contém 400 caracteres, usando como semente o trecho "Camargo era pouco simpático à primeira vista. Tinha as feições duras e frias, os olhos perscrutadores e sagazes, de uma sagacidade incômoda para quem encarava com eles, o que o não fazia atraente."da obra Helena, de Machado de Assis (domínio público) - a temperatura escolhida para a geração dos textos foi 0.5. Conforme explicado anteriormente, também será utilizado um gráfico para fazer a análise quantitativa.

No texto gerado neste experimento pode-se notar que a rede aprendeu várias palavras do português e as posiciona, normalmente, de acordo com a norma culta, respeitando regras mais simples como usar letra maiúscula depois de ponto. Apesar disto, há falta de sentido geral no texto. No gráfico, o erro de treino mantém-se maior que o de teste no inicio, podendo-se atribuir isto ao fato de que o erro do treino é calculado com o número de entradas igual ao *batch size*, 128, já o de teste usa 20% do total, equivalente a aproximadamente 63 mil entradas. Pode-se notar também que, depois da época 17, o erro de teste começa a variar pouco enquanto o de

Camargo era pouco simpático à primeira vista. Tinha as feições duras e frias, os olhos perscrutadores e sagazes, de uma sagacidade incômoda para quem encarava com eles, o que o não fazia atraente. Estava de depois, mas a morte de pernos de contratos. Eu ou ou dome; mas não não teria o despero de admiração e do senhor, a mesma alguma nossa alguma de lava um primeiro pergunto acabar o lado do caso de Santa Marcela, que não se não descontado na casa do Palha nem de mesmo e a casa da indercação de casa a intenção de grande outra casa da cansada. Era uma vida da menaria de contente de parte do

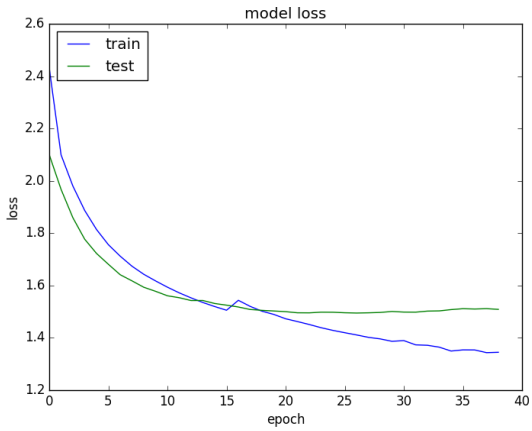

Figura 9: Gráfico contendo erro de treino e erro de teste, resultante do experimento 3.

treino só diminui, ocorrendo então um *overfit*.

3.1 Experimento 1

> Camargo era pouco simpático à primeira vista. Tinha as feições duras e frias, os olhos perscrutadores e sagazes, de uma sagacidade incômoda para quem encarava com eles, o que o não fazia atraente. Com meio mais para consciência de ambos. Adeundo sobrevi a casa que se antes de mim este menos que não me falta a filha, um papel de amor, e o valer de calas do companho de canto, como as carças da manhã sentado. Era um converso em verdade, atanhou-se na casa. A princípio de tal vezes me acharam a obrigado e os olhos do meu casamer, ou a porta de algumas portas e a resporta de alguma toacar a alm

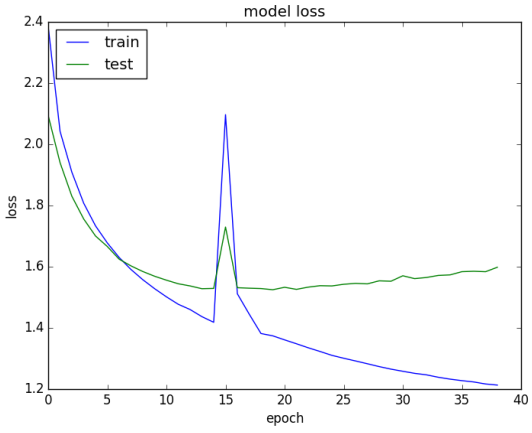

Figura 10: Gráfico contendo erro de treino e erro de teste, resultante do experimento 1.

No experimento 1, conforme o gráfico à cima, o *overfit* começa a ocorrer por volta da época 6, mantendo o erro do grupo de teste superior ao do experimento 3. Com isto, pode-se notar a influência do *dropout* para combater o *overfit*, conforme esperado segundo a literatura porém o erro mínimo do experimento 1 é menor.

3.2 Experimento 2

> Camargo era pouco simpático à primeira vista. Tinha as feições duras e frias, os olhos perscrutadores e sagazes, de uma sagacidade incômoda para quem encarava com eles, o que o não fazia atraente. Com meio mais para consciência de ambos. Adeundo sobrevi a casa que se antes de mim este menos que não me falta a filha, um papel de amor, e o valer de calas do companho de canto, como as carças da manhã sentado. Era um converso em verdade, atanhou-se na casa. A princípio de tal vezes me acharam a obrigado e os olhos do meu casamer, ou a porta de algumas portas e a resporta de alguma toacar a alm

De acordo com o gráfico, no experimento 2 o *overfit* começou a ocorrer por volta da época 32 porém ao comprar os erros mínimos, no experimento 2 o menor erro foi de 1.4946, já no experimento 3 foi de 1.4948. Em adição, o erro de treino do experimento 3 foi de 1.41, muito menor que o de teste, ou seja, estava em uma região de *overfit*.

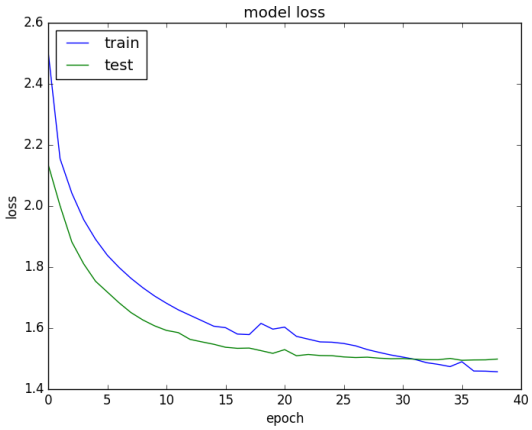

Figura 11: Gráfico contendo erro de treino e erro de teste, resultante do experimento 2.

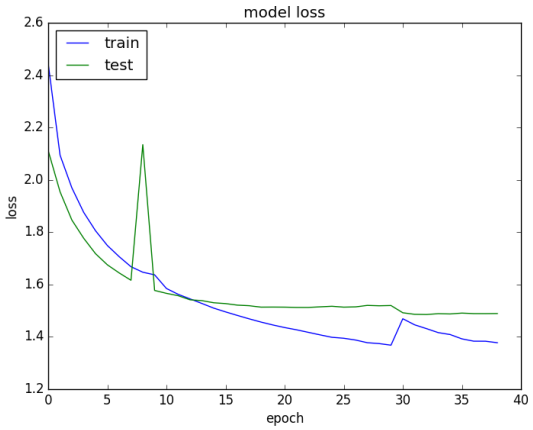

Figura 12: Gráfico contendo erro de treino e erro de teste, resultante do experimento 4.

> Camargo era pouco simpático à primeira vista. Tinha as feições duras e frias, os olhos perscrutadores e sagazes, de uma sagacidade incômoda para quem encarava com eles, o que o não fazia atraente. As de casa de sem nada. Estava outra vez em tentei esta passar de minha farada do arme, a primeira carta feita de pensar a mão do simples diante da minha cara do partico de meu amor. Talvez que era medo de moça com a espera e de uma contempação de perguntou a palavra, na casa de mas a mesma tora na bela nossa consciência dos portas e esperar a mão de confundidade, e a carta era morrer ao menos co

3.3 Experimento 4

Neste experimento, quantitativamente, destaca-se um pico durante o teste na época 8, indo de 1.6 para 2.1, voltando para 1.5 na próxima época, tal fato possivelmente ocorreu devido ao *dropout* da rede, que ocasiona esses picos durante o processamento do *Adam*. Vale notar que o erro mínimo para este experimento ocorre depois de começar a ter *overfitting*. Qualitativamente possui apenas uma palavra escrita errada a mais do que o experimento 3 mas não apresenta melhora significativa na coesão.

3.4 Experimento 5

Comparando os gráficos dos experimentos 3, 4 e 5 e os respectivos erros mínimos, 1.49, 1.48 e 1.51, pode-se concluir que, das três opções, a janela de tamanho 60 é a que produz o menor erro. Nota-se também que neste experimento o erro do grupo de teste para de diminuir na época 19 e que na época 38 o erro aumenta consideravelmente.

Camargo era pouco simpático à primeira vista. Tinha as feições duras e frias, os olhos perscrutadores e sagazes, de uma sagacidade incômoda para quem encarava com eles, o que o não fazia atraente. Como é para a mulher. A mulher era a mesma que era teria carros estavam as paras de Deus, e a dar por se fiz recura que eu tinha a minha invenção de esperando de falta, e atentou o mais de para deixar a porta de minha mãe do um compo que o seria este lembrando a parte dos homens no senamor, e provavelmente a minha senhora, fora diante mais que fizera de assim para a mesma vez e do seminário, algu

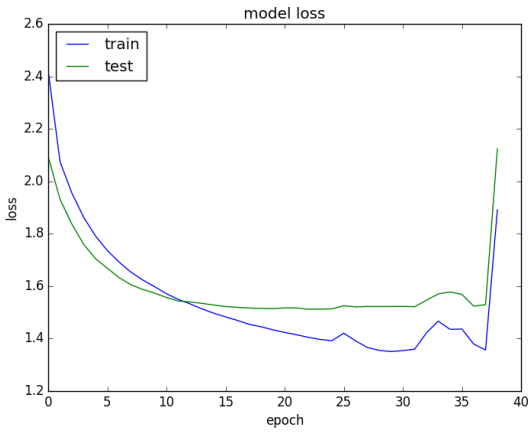

Figura 13: Gráfico contendo erro de treino e erro de teste, resultante do experimento 5.

3.5 Experimento 6

O erro mínimo de teste deste experimento é 1.54 enquanto o do experimento 3 é de 1.49, implicando que pelo menos quantitativamente não é melhor. Qualitativamente, o texto gerado já

Camargo era pouco simpático à primeira vista. Tinha as feições duras e frias, os olhos perscrutadores e sagazes, de uma sagacidade incômoda para quem encarava com eles, o que o não fazia atraente. Estava a composta de ao capítulo e a palavra não sei que ele estivosse do carto e o que entreia ter o meu ponto em minha mãe, e que não ter o tero esparto de confissar o destro do mais andando de contrapandeira que eu tenho de um experimento de trativamento da minha mais do mar de os almadeiros de casa da malher. Esteva a minha porta da mesma gesto. A gosta estava para o levado da filha, que me e

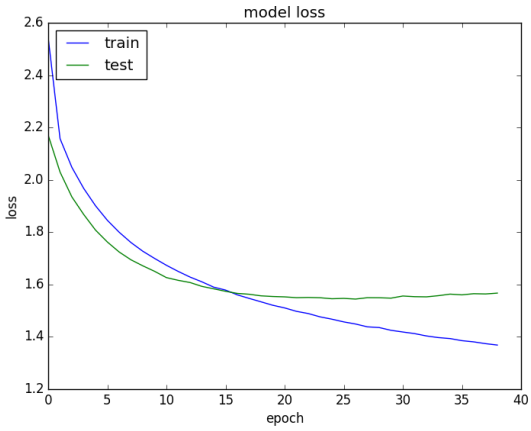

Figura 14: Gráfico contendo erro de treino e erro de teste, resultante do experimento 6.

não é coeso, além de conter mais palavras erradas.

> Camargo era pouco simpático à primeira vista. Tinha as feições duras e frias, os olhos perscrutadores e sagazes, de uma sagacidade incômoda para quem encarava com eles, o que o não fazia atraente. Eram os braços, acabavam da aquilado da melhor de filha do estado, dessas para se não disse entrou de portar delícios. O amor de contesso de atençar o minhe de lutar com a sem nada. Em verdade é a alguns presentes de um estar de natureza a manhã, que contecias a desperar de contar o brince, e de mim a minha felas e não persinas de a contessira de algumas suas algumas coisas e detrados de um modo

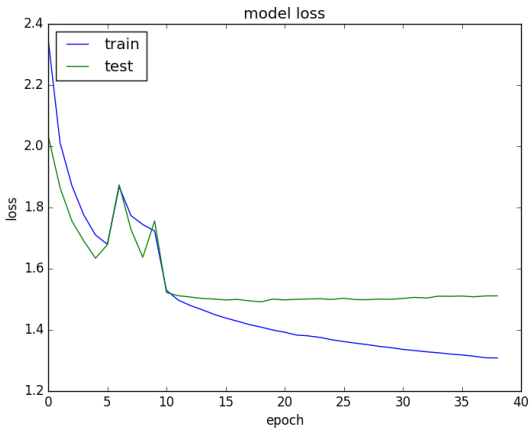

Figura 15: Gráfico contendo erro de treino e erro de teste, resultante do experimento 7.

3.6 Experimento 7

O erro mínimo deste experimento é próximo ao erro do experimento 3. Observando o gráfico pode-se notar que o erro de teste se estabiliza na época 10, enquanto no experimento 3 isto ocorre na época 22, este fato é explicável uma vez que para o mesmo número de épocas o experimento 7 terá o dobro de atualizações do gradiente, visto que a *batch size* é a metade, apesar disto apresenta picos maiores do que no experimento 3. Para fins de análise qualitativa, pode-se ressaltar o maior número de frases erradas e menor coesão.

3.7 Experimento 8

> Camargo era pouco simpático à primeira vista. Tinha as feições duras e frias, os olhos perscrutadores e sagazes, de uma sagacidade incômoda para quem encarava com eles, o que o não fazia atraente. Era para que estava sentido de alguns instantes. Não é que o proposso ficaram o pai de Sancha fara de largo, de casar um palseca de sentimentos. Estava anos, e as desconas era como este disconte e a minha filha. Não não achou a infinição do famo de lado e contrara a proteria de uma calar a um papel de algum tempo. Estava de dizer que ela estava em minha mãe de resposta, e a parte de meu casamento

No gráfico deste experimento pode-se notar que começa a ocorrer *overfitting* a partir da época 9, de tal forma que mesmo tendo um baixo erro de treino, comparado ao experimento 3, o erro de teste não diminui tanto, apesar de ser menor que o dos demais experimentos exceto o 1, fato explicável por conter mais neurônios. O texto gerado possui mais coesão porém tem mais palavras erradas, possivelmente por ter passado apenas por 13 épocas para chegar no erro de teste mínimo.

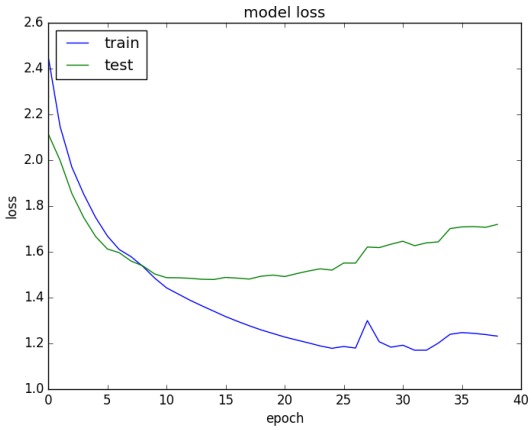

Figura 16: Gráfico contendo erro de treino e erro de teste, resultante do experimento 8.

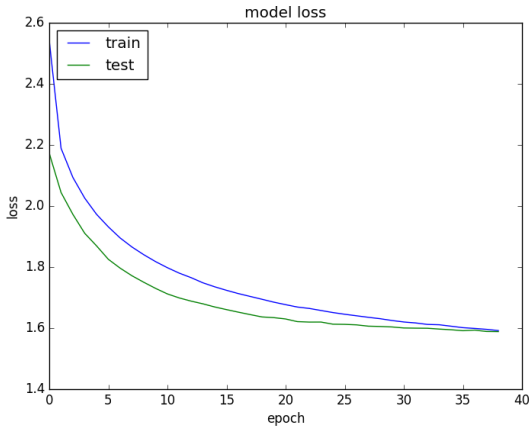

Figura 17: Gráfico contendo erro de treino e erro de teste, resultante do experimento 9.

3.8 Experimento 9

> Camargo era pouco simpático à primeira vista. Tinha as feições duras e frias, os olhos perscrutadores e sagazes, de uma sagacidade incômoda para quem encarava com eles, o que o não fazia atraente.
> - Eu desse interiala de falar se por de nem cara da manhã perdure da casa voltar que eu não não podia a dera não alguma vez de merento de um palegor a parte e de meio de estar da esporada e assim que contentava o menos faso, pretina perguntar-se a pronto nem com esta filha. A minha mãe levara e a distorço de algum trosto que constretou a também dar de contrarendo, o parreiro e verdadeira com a a

Conforme visto na tabela, neste experimento há metade dos neurônios do terceiro experimento, comprando os dois gráficos pode-se notar que o erro mínimo tanto de teste quanto no de treino e que no mesmo número de épocas a diferença entre o erro de teste e de treino só diminui no final. Em relação ao texto, pode-se notar um maior número de palavras não presentes na língua portuguesa.

4 Conclusão

Dessa forma, fica explicito a grande utilidade das ferramentas de *Deep Learning* e de Redes Neurais para trabalhos envolvendo análise e classificação de um grande número de dados, bem como a importância do LSTM para um aperfeiçoamento do resultado. Destaca-se ainda, o uso dessas técnicas no problema de geração automática de texto e o impacto causado por variáveis como o tamanho da janela, o *batch size* e o *dropout*.

A partir dos experimentos apresentados é possível dizer que uma rede com os parâmetros:

Tamanho de janela: 60

Neurônios: 512
Dropout: 0.2
Batch size: 128
Apresentaria os melhores resultados.

Ainda há muito a ser explorado no meio do *Deep Learning* com redes neurais recorrentes. Uma possível extensão do trabalho apresentado é o uso dessa ferramente para a geração de músicas, ao invés de textos. Nesse caso a entrada seria composta não mais de caracteres e sim de notas musicais, e a saída uma melodia formada pela composição dessas notas. A melhor adaptação do modelo, então, seria caraterizada pela harmonia da música no lugar da coerência e gramática do texto.

5 Bibliografia

Referências

[1] L. Bottou, *Large-scale machine learning with stochastic gradient descent*, in Proceedings of COMPSTAT'2010, Springer, 2010, pp. 177–186.

[2] D. Eck and J. Schmidhuber, *Finding temporal structure in music: blues improvisation with lstm recurrent networks*, in Proceedings of the 12th IEEE Workshop on Neural Networks for Signal Processing, 2002, pp. 747–756.

[3] D. D. S. GOMES, *Inteligência artificial: Conceitos e aplicações*, Olhar Científico, 1 (2011), pp. 234–246.

[4] A. Graves, *Generating sequences with recurrent neural networks*, arXiv preprint arXiv:1308.0850, (2013).

[5] S. Hochreiter and J. Schmidhuber, *Long short-term memory*, Neural Computation, 9 (1997), pp. 1735–1780.

[6] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, *Labeled faces in the wild: A database for studying face recognition in unconstrained environments*, Tech. Rep. 07-49, University of Massachusetts, Amherst, October 2007.

[7] A. Karpathy and L. Fei-Fei, *Deep visual-semantic alignments for generating image descriptions*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3128–3137.

[8] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, CoRR, abs/1412.6980 (2014).

[9] I. Sutskever, O. Vinyals, and Q. V. Le, *Sequence to sequence learning with neural networks*, in Advances in Neural Information Processing Systems 27, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, eds., Curran Associates, Inc., 2014, pp. 3104–3112.

[10] M. D. Zeiler, *Adadelta: an adaptive learning rate method*, arXiv preprint arXiv:1212.5701, (2012).