Programa Interdisciplinar de Pós-Graduação em

Computação Aplicada

Mestrado Acadêmico

Ingo Jost

Aplicação de *Deep Learning* em Dados Refinados para Mineração de Opiniões

São Leopoldo, 2015

UNIVERSIDADE DO VALE DO RIO DOS SINOS — UNISINOS
UNIDADE ACADÊMICA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA
NÍVEL MESTRADO

INGO JOST

APLICAÇÃO DE *DEEP LEARNING* EM DADOS REFINADOS PARA MINERAÇÃO DE OPINIÕES

SÃO LEOPOLDO
2015

Ingo Jost

APLICAÇÃO DE *DEEP LEARNING* EM DADOS REFINADOS PARA MINERAÇÃO DE OPINIÕES

Dissertação apresentada como requisito parcial para a obtenção do título de Mestre pelo Programa de Pós-Graduação em Computação Aplicada da Universidade do Vale do Rio dos Sinos — UNISINOS

Orientador:
Prof. Dr. João F. Valiati

São Leopoldo
2015

DADOS INTERNACIONAIS DE CATALOGAÇÃO NA PUBLICAÇÃO (CIP)

Jost, Ingo

Aplicação de *Deep Learning* em dados refinados para Mineração de Opiniões / Ingo Jost — 2015.

88 f.: il.; 30 cm.

Dissertação (mestrado) — Universidade do Vale do Rio dos Sinos, Programa de Pós-Graduação em Computação Aplicada, São Leopoldo, 2015.

"Orientador: Prof. Dr. João F. Valiati, Unidade Acadêmica de Pesquisa e Pós-Graduação".

1. *Deep Learning*. 2. Mineração de Opiniões. 3. *Deep Belief Networks*. I. Título.

CDU 004.9

Bibliotecário responsável: Flávio Nunes — CRB 10/1298

AGRADECIMENTOS

Gostaria de agradecer a todos que de alguma maneira contribuíram para a realização desse trabalho: aos amigos que sempre pude contar, à minha família, à minha esposa Mari: minha gratidão pelo apoio. Agradeço também aos colegas e corpo docente do PIPCA, em especial ao professor Dr. João F. Valiati, pela dedicação, conhecimentos e orientação.

"*Se você não mudar a direção, terminará no mesmo lugar de onde partiu*".
(Provérbio chinês)

RESUMO

Deep Learning é uma sub-área de Aprendizado de Máquina que tem obtido resultados satisfatórios em várias áreas de aplicação, implementada por diferentes algoritmos, como *Stacked Auto-encoders* ou *Deep Belief Networks*. Este trabalho propõe uma modelagem que aplica uma implementação de um classificador que aborda técnicas de *Deep Learning* em Mineração de Opiniões, área que tem sido alvo de constantes estudos, dada a necessidade das corporações buscarem a compreensão que clientes possuem de seus produtos ou serviços. O favorecimento do crescimento de Mineração de Opiniões também se dá pelo ambiente colaborativo da Web 2.0, em que várias ferramentas propiciam a emissão de opiniões. Os dados utilizados passaram por um refinamento na etapa de pré-processamento com o intuito de aplicar *Deep Learning*, da qual uma das principais atribuições é a seleção de características, em dados refinados em vez de dados mais brutos. A promissora tecnologia de *Deep Learning* combinada com a estratégia de refinamento demonstrou nos experimentos a obtenção de resultados competitivos com outros estudos relacionados e abrem perspectiva de extensão deste trabalho.

Palavras-chave: *Deep Learning*. Mineração de Opiniões. *Deep Belief Networks*.

ABSTRACT

Deep Learning is a Machine Learning's sub-area that have achieved satisfactory results in different application areas, implemented by different algorithms, such as Stacked Auto-encoders or Deep Belief Networks. This work proposes a research that applies a classifier that implements Deep Learning concepts in Opinion Mining, area has been approached by constant researches, due the need of corporations seeking the understanding that customers have of your products or services. The Opinion Mining's growth is favored also by the collaborative Web 2.0 environment, where multiple tools provide issuing opinions. The data used for experiments were refined in preprocessing step in order to apply Deep Learning, which it one of the main tasks the feature selection, in refined data, instead of applying Deep Learning in more raw data. The refinement strategy combined with the promising technology of Deep Learning has demonstrated in preliminary experiments the achievement of competitive results with other studies and opens the perspective for extension of this work.

Keywords: Deep Learning. Opinion Mining. Deep Belief Networks.

LISTA DE FIGURAS

Figura 1: Etapas do processo de KDD 26
Figura 2: Estrutura de uma Rede Neural MLP 30
Figura 3: Estrutura de um neurônio artificial 31
Figura 4: Estrutura de uma *Boltzmann Machine* 34
Figura 5: Estrutura de uma RBM 35
Figura 6: *Pre-training* e *Fine-tuning* 37
Figura 7: Diferença entre RBM (A) e CRBM (B) 37

Figura 8: Exemplo de amostras para ilustrar densidade de informações 42
Figura 9: Comparação de resultados ADN x IADN, com 10 amostras rotuladas 43
Figura 10: Gráfico das funções *rectifier* e *softplus* 44
Figura 11: *Transfer loss* para os diferentes domínios, comparando diferentes técnicas . 46

Figura 12: Estrutura da modelagem 50
Figura 13: Tempo de execução - 120 (a) e 160 (b) épocas *Pre-training* e *Fine-tuning* . . 56
Figura 14: *Recall* (%) p/ filmes com 120 (a) e 160 (b) épocas *Pre-training* e *Fine-tuning* 57
Figura 15: Precisão (%) p/ filmes com 120 (a) e 160 (b) épocas *Pre-training* e *Fine-tuning* 57
Figura 16: *Recall* (%) p/ GPS com 120 (a) e 160 (b) épocas *Pre-training* e *Fine-tuning* . 57
Figura 17: Precisão (%) p/ GPS com 120 (a) e 160 (b) épocas *Pre-training* e *Fine-tuning* 58
Figura 18: *Recall* (%) p/ câmeras com 120 (a) e 160 (b) épocas *Pre-training* e *Fine-tuning* 58
Figura 19: Precisão (%) p/ câmeras com 120 (a) e 160 (b) épocas *Pre-training* e *Fine-tuning* 58
Figura 20: *Recall* (%) p/ livros com 120 (a) e 160 (b) épocas *Pre-training* e *Fine-tuning* 59
Figura 21: Precisão (%) p/ livros com 120 (a) e 160 (b) épocas *Pre-training* e *Fine-tuning* 59
Figura 22: *F-measure* (%) p/ filmes com 120 (a) e 160 (b) épocas *Pre-training* e *Fine-tuning* 60
Figura 23: *F-measure* (%) p/ GPS com 120 (a) e 160 (b) épocas *Pre-training* e *Fine-tuning* 60
Figura 24: *F-measure* (%) p/ câmeras com 120 (a) e 160 (b) épocas *Pre-training* e *Fine-tuning* 60
Figura 25: *F-measure* (%) p/ livros com 120 (a) e 160 (b) épocas *Pre-training* e *Fine-tuning* 61
Figura 26: Variação das melhores configurações 63
Figura 27: Tempo de execução - 30 épocas *Pre-training* e 120 *Fine-tuning* 64

LISTA DE TABELAS

Tabela 1: Matriz de Confusão 38

Tabela 2: Acurácia (%) obtida pelo modelo de *Active Deep Network* e DBN 42
Tabela 3: Acurácia (%) obtida pelo modelo IADN, com 100 amostras rotuladas..... 43
Tabela 4: Erro de teste (%) na aplicação das diferentes funções..... 44

Tabela 5: Acurácia (%) - filmes - 120 épocas *Pre-training* e *Fine-tuning*..... 53
Tabela 6: Acurácia (%) - GPS - 120 épocas *Pre-training* e *Fine-tuning* 54
Tabela 7: Acurácia (%) - câmeras - 120 épocas *Pre-training* e *Fine-tuning* 54
Tabela 8: Acurácia (%) - livros - 120 épocas *Pre-training* e *Fine-tuning* 54
Tabela 9: Acurácia (%) - filmes - 160 épocas *Pre-training* e *Fine-tuning*..... 55
Tabela 10: Acurácia (%) - câmeras - 160 épocas *Pre-training* e *Fine-tuning* 55
Tabela 11: Acurácia (%) - livros - 160 épocas *Pre-training* e *Fine-tuning* 55
Tabela 12: Acurácia (%) - GPS - 160 épocas *Pre-training* e *Fine-tuning* 55
Tabela 13: Acurácia (%) - filmes - 30 épocas *Pre-training* e 120 épocas *Fine-tuning* . . 62
Tabela 14: Acurácia (%) - GPS - 30 épocas *Pre-training* e 120 épocas *Fine-tuning* . . . 62
Tabela 15: Acurácia (%) - câmeras - 30 épocas *Pre-training* e 120 épocas *Fine-tuning* . 62
Tabela 16: Acurácia (%) - livros - 30 épocas *Pre-training* e 120 épocas *Fine-tuning* . . 62
Tabela 17: Desv.padr. - Acurácia (%) - filmes - 30 épocas *Pre-training* e 120 épocas *Fine-tuning*..... 64
Tabela 18: Desv.padr. - Minutos de Exec. - filmes - 30 épocas *Pre-training* e 120 épocas *Fine-tuning*..... 64
Tabela 19: Acurácia (%) - Comparação com trabalhos relacionados..... 65
Tabela 20: Acurácia (%) - Comparação com trabalho Moraes (2013)..... 65

Tabela 21: Precisão (%) POS - GPS - 120 épocas *Pre-training* e *Fine-tuning*..... 77
Tabela 22: Precisão (%) NEG - GPS - 120 épocas *Pre-training* e *Fine-tuning*..... 77
Tabela 23: Precisão (%) POS - GPS - 160 épocas *Pre-training* e *Fine-tuning*..... 77
Tabela 24: Precisão (%) NEG - GPS - 160 épocas *Pre-training* e *Fine-tuning*..... 78
Tabela 25: Precisão (%) POS - câmeras - 120 épocas *Pre-training* e *Fine-tuning* 78
Tabela 26: Precisão (%) NEG - câmeras - 120 épocas *Pre-training* e *Fine-tuning* 78
Tabela 27: Precisão (%) POS - câmeras - 160 épocas *Pre-training* e *Fine-tuning* 78
Tabela 28: Precisão (%) NEG - câmeras - 160 épocas *Pre-training* e *Fine-tuning* 78
Tabela 29: Precisão (%) POS - livros - 120 épocas *Pre-training* e *Fine-tuning* 79
Tabela 30: Precisão (%) NEG - livros - 120 épocas *Pre-training* e *Fine-tuning* 79
Tabela 31: Precisão (%) POS - livros - 160 épocas *Pre-training* e *Fine-tuning* 79
Tabela 32: Precisão (%) NEG - livros - 160 épocas *Pre-training* e *Fine-tuning* 79
Tabela 33: Precisão (%) POS - filmes - 120 épocas *Pre-training* e *Fine-tuning* 79
Tabela 34: Precisão (%) NEG - filmes - 120 épocas *Pre-training* e *Fine-tuning* 80
Tabela 35: Precisão (%) POS - filmes - 160 épocas *Pre-training* e *Fine-tuning* 80
Tabela 36: Precisão (%) NEG - filmes - 160 épocas *Pre-training* e *Fine-tuning* 80

Tabela 37: *F-measure* (%) - GPS - 120 épocas *Pre-training* e *Fine-tuning* 81
Tabela 38: *F-measure* (%) - GPS - 160 épocas *Pre-training* e *Fine-tuning* 81
Tabela 39: *F-measure* (%) - câmeras - 120 épocas *Pre-training* e *Fine-tuning*..... 81
Tabela 40: *F-measure* (%) - câmeras - 160 épocas *Pre-training* e *Fine-tuning*..... 82
Tabela 41: *F-measure* (%) - livros - 120 épocas *Pre-training* e *Fine-tuning*..... 82
Tabela 42: *F-measure* (%) - livros - 160 épocas *Pre-training* e *Fine-tuning*..... 82

Tabela 43: *F-measure* (%) - filmes - 120 épocas *Pre-training* e *Fine-tuning*.....82
Tabela 44: *F-measure* (%) - filmes - 160 épocas *Pre-training* e *Fine-tuning*.....82

Tabela 45: *Recall* (%) POS - GPS -120 épocas *Pre-training* e *Fine-tuning*.....83
Tabela 46: *Recall* (%) NEG - GPS - 120 épocas *Pre-training* e *Fine-tuning*.....83
Tabela 47: *Recall* (%) POS - GPS - 160 épocas *Pre-training* e *Fine-tuning*.....83
Tabela 48: *Recall* (%) NEG - GPS - 160 épocas *Pre-training* e *Fine-tuning*.....84
Tabela 49: *Recall* (%) POS - câmeras - 120 épocas *Pre-training* e *Fine-tuning* 84
Tabela 50: *Recall* (%) NEG - câmeras - 120 épocas *Pre-training* e *Fine-tuning* 84
Tabela 51: *Recall* (%) POS - câmeras - 160 épocas *Pre-training* e *Fine-tuning* 84
Tabela 52: *Recall* (%) NEG - câmeras - 160 épocas *Pre-training* e *Fine-tuning* 84
Tabela 53: *Recall* (%) POS - livros - 120 épocas *Pre-training* e *Fine-tuning* 85
Tabela 54: *Recall* (%) NEG - livros - 120 épocas *Pre-training* e *Fine-tuning* 85
Tabela 55: *Recall* (%) POS - livros - 160 épocas *Pre-training* e *Fine-tuning* 85
Tabela 56: *Recall* (%) NEG - livros - 160 épocas *Pre-training* e *Fine-tuning* 85
Tabela 57: *Recall* (%) POS - filmes - 120 épocas *Pre-training* e *Fine-tuning* 85
Tabela 58: *Recall* (%) NEG - filmes - 120 épocas *Pre-training* e *Fine-tuning* 86
Tabela 59: *Recall* (%) POS - filmes - 160 épocas *Pre-training* e *Fine-tuning* 86
Tabela 60: *Recall* (%) NEG - filmes - 160 épocas *Pre-training* e *Fine-tuning* 86

LISTA DE ABREVIATURAS

Pres. Trab.	Presente Trabalho, -s
Desv.padr.	Desvio-padrão
Exec.	Execução
Alc. API	*AlchemyAPI*

LISTA DE SIGLAS

ADN	*Active Deep Network*
BOW	*bag-of-words*
CAE	*Convolutional Auto-encoders*
CD	*Contrastive Divergence*
CNN	*Convolutional Neural Networks*
CRBM	*Conditional Restricted Boltzmann Machines*
DBN	*Deep Belief Networks*
DBM	*Deep Boltzmann Machines*
EBM	*Energy-Based Model*
EQM	Erro Quadrático Médio
IADN	*Information Active Deep Network*
ICA	*Independent Component Analysis*
IG	*Information Gain*
KDD	*Knowledge Discovery in Database*
MCT	*Multi-label Consensus Training*
MLP	*Multilayer Perceptron*
NN	*Neural Networks*
NLP	*Natural Language Processing*
PCA	*Principal Component Analysis*
RBM	*Restricted Boltzmann Machines*
RNA	Redes Neurais Artificiais
RNTN	*Recursive Neural Tensor Network*
RNN	*Rectifier Neural Network*
SAE	*Stacked Auto-encoder*
SCL	*Structural Correspondence Learning*
SDA	*Stacked Denoised Auto-encoder*
SFA	*Spectral Feature Alignment*
SVM	*Support Vector Machine*
TF-IDF	*Term Frequency-Inverse Document Frequency*
VSM	*Vector Space Model*

SUMÁRIO

1 INTRODUÇÃO 21
1.1 Objetivo 22
1.2 Estrutura 23

2 FUNDAMENTAÇÃO TEÓRICA 25
2.1 Mineração de Dados 25
2.2 Mineração de Opiniões 26
2.2.1 Pré-processamento Textual 27
2.2.2 Seleção de Termos 28
2.2.3 *Vector Space Model* 29
2.3 Redes Neurais Artificiais 29
2.4 *Deep Learning* 33
2.4.1 *Deep Belief Networks* 34
2.5 Métricas para Análise de Resultados 37
2.6 Considerações 39

3 TRABALHOS RELACIONADOS 41
3.1 *Active Deep Network* 41
3.2 Rede Neural Retificadora 43
3.3 Adaptação de Domínio 45
3.4 Considerações 46

4 MODELAGEM PROPOSTA E RESULTADOS OBTIDOS 49
4.1 Metodologia 49
4.1.1 Análise dos Resultados 51
4.2 Experimentos Computacionais 51
4.2.1 Dados Utilizados 51
4.2.2 Ferramentas 52
4.2.3 Implementação 52
4.2.4 Resultados 53
4.3 Considerações 64

5 CONCLUSÃO 67

REFERÊNCIAS 69

APÊNDICE A *ALGORITMO CONTRASTIVE DIVERGENCE* 75

APÊNDICE B VALORES DE PRECISÃO 77

APÊNDICE C VALORES DE *F-MEASURE* 81

APÊNDICE D VALORES DE *RECALL* 83

APÊNDICE E CÓDIGO-FONTE 87

1 INTRODUÇÃO

O contínuo crescimento do volume de dados gerados favoreceu o desenvolvimento de técnicas que buscam obter o conhecimento implícito nesses dados. Neste cenário destaca-se a Descoberta de Conhecimento (KDD, do inglês *Knowledge Discovery in Databases*), estudada há décadas e que continua sendo desenvolvida para descoberta de informações e identificação de padrões.

Seu desenvolvimento foi amparado também pelos avanços tecnológicos dos últimos anos, possibilitando sua atuação em diferentes campos de pesquisa. Esta vasta abrangência possibilita a evolução da grande área de Aprendizado de Máquina, através do aperfeiçoamento de técnicas existentes e pesquisa de novos conceitos e algoritmos (VINCENT et al., 2008).

Neste contexto surge *Deep Learning*, área que possui diferentes implementações que buscam identificar e selecionar características dos dados, possibilitando diferentes formas de representação destes (ARNOLD, 2013), reduzindo também sua dimensionalidade. Estudos têm demonstrado que através de sua aplicação são proporcionados melhores resultados para modelos que visam classificação de dados (JARRETT; KAVUKCUOGLU; LECUN, 2009), (YU; DENG; WANG, 2009).

Deep Learning é inspirada em uma das técnicas mais disseminadas de Aprendizado de Máquina, as Redes Neurais Artificiais, e tem sido alvo de estudos em distintas áreas: reconhecimento de imagens (NAIR; HINTON, 2010), áudio (NOROUZI, 2009), caracteres (HINTON; SALAKHUTDINOV, 2006) e reconhecimento facial (FAN et al., 2014). Não somente no meio acadêmico encontram-se aplicações fazendo uso de *Deep Learning*, como grandes corporações também têm dedicado esforços em projetos que a utilizam: *Google* [1], com carros que dispensam motoristas, óculos-computadores e algoritmos de seu motor de buscas; *Microsoft* [2], com o Projeto Adam, ferramenta que promete eficiência e rapidez no reconhecimento visual; e *Baidu* [3] que criou um laboratório de *Deep Learning* para seus projetos, que também incluem motor de buscas e carro inteligente.

Dado o sucesso já obtido em áreas complexas, juntamente com os grandes investimentos em estudos na área, tanto no meio acadêmico quanto no corporativo, abrem-se perspectivas de utilização de *Deep Learning* em outras linhas de pesquisa, como a classificação textual, que pode ser utilizada, por exemplo, em classificação de documentos. Esta área na qual se busca identificar padrões em textos é chamada de Mineração Textual, e pode ser dividida em diferentes especializações, como a voltada à análise de opiniões: *Opinion Mining* (Mineração de Opiniões e Análise de Sentimentos) (LIU, 2012). Esta distinção em uma nova área ocorre devido ao surgimento de várias questões inerentes a opiniões (PANG; LEE, 2008).

O crescimento desta especialização é favorecido pelo cenário atual da Web 2.0, um ambi-

[1] http://www.google.com
[2] http://www.microsoft.com/
[3] http://www.baidu.com/

ente colaborativo que possibilita o surgimento de ferramentas que propiciam (e estimulam) a emissão de opiniões pelos seus usuários: grupos de discussão, fóruns, redes sociais, blogs, sites de notícias, venda de produtos, entre outras. Soma-se a isto o fato das empresas buscarem cada vez mais descobrir a percepção de seus consumidores e (possíveis) clientes a respeito de determinados produtos ou serviços. Estas informações auxiliam na tomada de decisão e fortalecem a necessidade de estudos na área de Mineração de Opiniões.

Destaca-se ainda em Mineração de Opiniões que o objetivo de grande parte dos estudos é a identificação da polaridade, consistindo em reconhecer se determinada opinião pertence à classe *positiva* ou *negativa* (LIU, 2012). Para esta classificação já surgem mais questões a serem tratadas, por exemplo, o fato de textos muito distintos possuírem um mesmo significado (positivo ou negativo).

A abordagem de *Deep Learning* em Mineração de Opiniões já foi empregada em adaptação de domínio, como por Glorot, Bordes e Bengio (2011b), que coletaram opiniões de diferentes produtos, como brinquedos, câmeras, livros, eletrônicos e celulares, sendo que o modelo foi proposto para classificar opiniões de qualquer produto. Outros trabalhos que tratam a questão da polaridade já fizeram uso de *Deep Learning*, como em Zhou, Chen e Wang (2010) e Glorot, Bordes e Bengio (2011a), onde foram elaborados modelos com o objetivo de identificar os sentimentos presentes nas opiniões, classificando-as como positivas ou negativas.

Estes trabalhos apresentam a aplicação de *Deep Learning* em bases de dados amplamente exploradas, como Pang e Lee (2004), fazendo uso de *Deep Learning* para seleção de características, mas sem abordar procedimentos que poderiam melhorar os resultados, como o refinamento dos dados com o emprego de métodos de filtragem e redução da dimensionalidade.

Enquanto que as técnicas de *Deep Learning* pregam seu uso em informações mais brutas a fim de identificar automaticamente características, esse estudo pretende investigar o impacto da utilização de informações já refinadas para averiguar se uma pré-seleção de características poderia favorecer ainda mais o poder das técnicas de *Deep Learning*.

1.1 Objetivo

O objetivo do trabalho proposto é realizar uma investigação do impacto do uso de dados refinados sobre o problema de classificação de polaridade de opiniões textuais da web, assim como encontrado no trabalho de Moraes (2013), quando aplicados classificadores baseados em técnicas de *Deep Learning*, no sentido de produzir uma avaliação comparativa e crítica com a literatura corrente.

Tem-se ainda como contribuições:

- A obtenção de melhores resultados em relação aos trabalhos correlatos;
- Apresentação e demonstração de uma experimentação mais abrangente que a encontrada na literatura corrente, com o emprego de uma sistemática de variabilidade de parâmetros

e avaliação de métricas mais robustas do que a simples acurácia;

- Fornecer uma análise do papel que exerce o refinamento prévio de dados textuais para a aplicação de *Deep Learning*, visto que esta técnica por si só já possui a atribuição de extração de características;
- Apresentação de estudo comparativo com trabalhos relacionados como *benchmark*, visando averiguá-los para posterior análise crítica, comparando-os com os resultados dos experimentos deste trabalho;
- Produção de uma análise crítica e recomendação da utilização de *Deep Learning* em dados refinados.

1.2 Estrutura

O capítulo seguinte apresenta a fundamentação teórica referente à Mineração de Opiniões e tópicos referentes à Mineração Textual, além de abordar conceitos fundamentais para o entendimento de *Deep Learning* e suas implementações: Redes Neurais Artificiais (RNA) e *Deep Belief Networks* (DBN). No Capítulo 3 são abordados trabalhos relacionados com a aplicação de *Deep Learning* ligados à problemática do trabalho. A modelagem proposta juntamente com os experimentos e os resultados obtidos são apresentados no Capítulo 4. Por fim, conclusões e perspectiva de extensão ao trabalho são apresentados no Capítulo 5.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são expostos os principais conceitos abordados no presente trabalho. É discutido o conceito de Mineração de Dados, servindo como base para Mineração de Opiniões, além de serem tratadas questões inerentes à Mineração Textual.

São apresentadas as Redes Neurais Artificiais, que serviram de inspiração para *Deep Learning*, abordada no restante do capítulo, desde seus conceitos, até a sua implementação por parte das *Deep Belief Networks*. Por fim, são ainda discutidas as métricas utilizadas para validar os resultados obtidos.

2.1 Mineração de Dados

O processo de KDD é empregado para extração de padrões e tendências em conjuntos de dados, possibilitando a obtenção de um conhecimento não explícito, que pode ser utilizado na tomada de decisões. Este processo é dividido, conforme Fayyad et al. (1996), em cinco etapas, como pode ser visto na Figura 1.

- Seleção: nesta etapa são selecionados os dados a serem considerados para o processo de Descoberta de Conhecimento, sendo descartados os dados irrelevantes ou os que tendem a possuir uma menor contribuição para processo de KDD. Esta seleção pode ser horizontal (onde é selecionado parte do conjunto de dados) ou vertical (quando são eliminados atributos que não serão considerados no processo de KDD) (GOLDSCHMIDT; PASSOS, 2005);

- Pré-processamento: etapa em que é efetuada a limpeza dos dados, eliminando inconsistências (como erros de cadastro) e tratamento de dados, como a grande ocorrência de valores nulos em determinados campos, por exemplo;

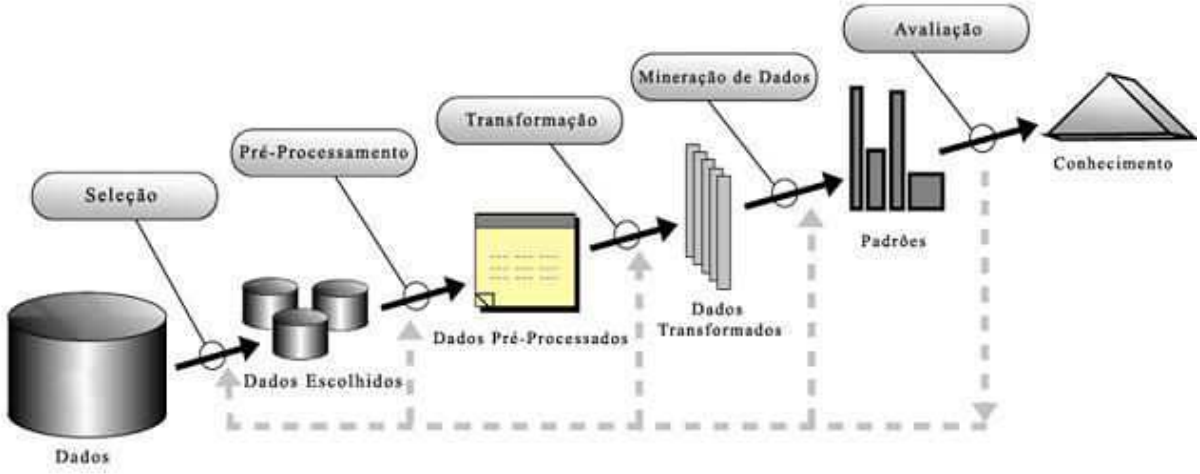
- Transformação: uniformização de dados com diferentes formatos mas com o mesmo significado (por serem oriundos de bases de dados distintas, por exemplo). Nesta etapa os dados são transformados no formato de entrada para as ferramentas de Mineração de Dados;

- Mineração de Dados: etapa na qual efetivamente ocorre a busca propriamente dita pelo conhecimento e identificação de padrões. Os algoritmos implementam diferentes técnicas, que podem ser adotadas de acordo com a demanda gerada pelo problema, podendo ser de clusterização, regras de associação ou classificação, por exemplo;

- Avaliação: onde ocorre a análise dos resultados obtidos para avaliação da importância do conhecimento adquirido.

Estas etapas fazem parte de um processo iterativo, podendo ser cíclicas, visando a obtenção de melhores resultados.

Figura 1: Etapas do processo de KDD



Fonte: Adaptado de Fayyad et al. (1996)

2.2 Mineração de Opiniões

A Mineração de Dados pode ser aplicada a dados textuais, neste caso conhecida como Mineração Textual, amplamente utilizada em diferentes problemas, como a classificação de opiniões. O grande número de particularidades envolvidas originou uma especialização a parte: Mineração de Opiniões, abordada neste trabalho.

Embora as opiniões sejam relacionadas a diferentes domínios (produtos ou notícias, por exemplo) e emitidas em diferentes ferramentas, estudos para a classificação de opiniões frequentemente são direcionados para um objetivo comum: identificação de polaridade (PANG; LEE, 2008). Esta finalidade consiste em verificar a ideia que a opinião transmite, geralmente classificando-a como *negativa* ou *positiva*.

Dentre as dificuldades da Mineração de Opiniões, destacam-se os diferentes formatos e tamanhos das opiniões, visto que para os seus emissores não há padronização. Por conta disso, opiniões com 130 ou 2500 caracteres sobre um mesmo produto podem ter o mesmo significado. Além disso, o emissor não possui compromisso com ortografia e concordância em seu texto, sendo livre a utilização de abreviações e gírias, em razão deste possuir, por vezes, um caráter de informalidade.

De acordo com Liu (2012), opiniões podem ser diferenciadas entre regulares (quando se referem a uma entidade, direta ou indiretamente) ou comparativas (mais de uma entidade é abordada) e a análise sobre textos opinativos ocorre em três níveis:

- Nível de Documento: identificação se todo o documento corresponde à classe *positiva* ou *negativa*;

- Nível de Sentença: análise de cada sentença do documento, pois estas podem transmitir diferentes ideias em relação à polaridade;

- Nível de Entidade e Aspectos: identificação de quais entidades são referidas pelas sentenças.

Muitos desafios também surgem tais como a subjetividade, em que os termos podem ter diferentes significados e uma opinião positiva pode possuir termos comumente utilizados para sentenças negativas; e negação, onde uma sequência de termos pode normalmente caracterizar uma opinião positiva, porém este mesmo conjunto pode representar o oposto, caso precedido de termos que indicam negação (PANG; LEE, 2008). Soma-se a estas questões a necessidade de tratamento da ocorrência de opiniões implícitas e condicionais e a presença de sentimentos difíceis de identificar, como ironia e sarcasmo, que junto com a sutileza empregada, podem prejudicar algoritmos de classificação.

Observa-se ainda a possibilidade de avaliação da intensidade das opiniões, onde em vez de apenas indicar se estas pertencem às classes *positiva* ou *negativa*, são atribuídos valores a elas (TANYILDIZI, 2009). Além das questões mencionadas acima, há ainda a necessidade de tratamento de problemas inerentes à área de Mineração Textual, incluindo alguns relacionados ao Processamento de Linguagem Natural (NLP, do inglês *Natural Language Processing*), referidas a seguir.

2.2.1 Pré-processamento Textual

Esta seção apresenta o pré-processamento, um dos passos mais importantes do processo de KDD (FAYYAD et al., 1996). Quando este é realizado sobre dados textuais, surgem problemas específicos, como grande quantidade de palavras, frases e sentenças (FELDMAN; SANGER, 2006). Para evitar problemas como estes, o volume de dados pode ser reduzido através de técnicas de seleção de termos, que podem vir a melhorar a qualidade dos dados.

Antes de realizar esta seleção, alguns métodos normalmente são aplicados: Tokenização, *Stemming* e Remoção de *Stopwords*.

2.2.1.1 Tokenização

Tokenização é a identificação de palavras (*tokens*) em um texto. Normalmente são verificados novos *tokens* em sinais de pontuação ou espaços em branco, sendo que ocorrem exceções, como por exemplo a contração do inglês *isn't*, composta por duas palavras (*is* e *not*) sem separação. Os espaços em branco podem também dividir um mesmo termo, como no caso em que *database* é escrito como *data base*.

Nesta etapa ainda é definido se serão utilizados *unigrams*, *bigrams* ou *trigrams*, correspondendo a sequências de uma, duas ou três palavras, respectivamente, a compor os *tokens*, sendo este modelo chamado de *bag-of-words* (BOW). Além da identificação dos *tokens*, há a análise do discurso (semântica), que conforme Santos (2002) consiste em encontrar relacionamentos entre as sentenças de um texto, como pronomes referenciando um mesmo sujeito.

2.2.1.2 *Stemming*

Algoritmos de *Stemming* permitem que termos diferentes, mas com o radical em comum, sejam agrupados em um mesmo termo. Este agrupamento pode ocorrer em verbos conjugados em diferentes tempos verbais ou substantivos com variações de singular e plural e substantivos primitivos e derivados, por exemplo.

Um dos algoritmos de *Stemming* mais utilizados é o de Porter (1997), presente em diversas ferramentas, como a *Text to Matrix Generator* (TMG), utilizada também na etapa de pré-processamento de Moraes (2013), mais detalhada no Capítulo 4.

2.2.1.3 Remoção de *Stopwords*

Algumas palavras podem ocorrer com grande frequência em textos, mas não chegam a contribuir para o processo de classificação ou identificação de padrões. Estas palavras podem ser removidas, diminuindo o volume de dados sem comprometer o conteúdo do texto.

Estas são comumente chamadas *stopwords*, presentes em listas já pré-definidas, bastando utilizá-las como parâmetro para a ferramenta de Mineração Textual para que estas sejam removidas. Outras estratégias podem ser adotadas, como criar manualmente uma lista de *stopwords*, caso seja um domínio bastante específico, ou como em Zhou, Chen e Wang (2010), que após obterem o conjunto de termos com suas frequências, simplesmente descartaram 1,5% dos termos mais frequentes na etapa de pré-processamento.

2.2.2 Seleção de Termos

Os procedimentos acima mencionados, juntamente com a etapa de seleção de termos, auxiliam na limpeza dos dados (redução do volume). Em um conceito geral, um termo pode ser uma palavra, símbolo ou número, dependendo do domínio da aplicação, e esta etapa almeja a redução do volume de dados e eliminação de termos irrelevantes (BERRY; KOGAN, 2010).

Diferentes técnicas podem ser empregadas para a seleção de termos (FORMAN, 2003), como *Probability Ratio* (PR), *Chi-squared* e *Information Gain* (IG) (MITCHELL, 1997), tendo sido esta última aplicada para o refinamento dos dados utilizados no presente trabalho. O procedimento faz uso do algoritmo Dividir para Conquistar, como Árvores de Decisão, onde é computada a entropia e calculado o ganho de informação para cada termo, aferindo um peso baseado na distribuição dos valores do termo em cada uma das classes, indicando o quanto cada termo pode determinar a qual classe a opinião pertence. O cálculo do IG é dado pela Equação (2.1)

$$IG(a_i) = \sum_{k=1}^{|C|} P(c_k) log_2 \frac{1}{P(c_k)} - \sum_{v=0}^{|V|} P(a_i = v) \sum_{k=1}^{|C|} P(c_k | a_i = v) log_2 \frac{1}{P(c_k | a_i = v)}, \quad (2.1)$$

onde, de acordo com Moraes (2013), $|C|$ corresponde ao total de classes e $|V|$ aos possíveis valores do atributo a_i; $P(c_k)$ é a probabilidade da ocorrência de uma amostra na classe c_k, $P(ai = v)$ é a probabilidade do atributo a_i ocorrer em uma amostra com o valor v, $P(c_k|a_i = v)$ é a probabilidade do atributo a_i ocorrer em uma amostra pertencente à classe c_k com o valor v .

2.2.3 *Vector Space Model*

Após a etapa de seleção de termos, os documentos e seus termos podem ser representados por *Vector Space Model* (VSM) (SALTON; MCGILL, 1986), no qual os documentos são vetores e cada dimensão destes é associada a um dos termos selecionados (BERRY; KOGAN, 2010). Esta técnica permite representar um documento através de valores armazenados nas posições dos vetores, possivelmente referenciando a frequência do termo, como no trabalho relacionado de Zhou, Chen e Wang (2010), ou seu *Term Frequency-Inverse Document Frequency* (TF-IDF) (PALTOGLOU; THELWALL, 2010).

O TF-IDF ocorre pela multiplicação de TF, Equação (2.2), na qual $t_i d_j$ é o número de ocorrências do termo t_i no documento d_j, e o denominador é a soma das ocorrências de todos os k termos do documento d_j; e IDF, Equação (2.3), onde o total de documentos D é dividido pela quantidade de documentos que contêm o termo t_i, aplicando-se o logaritmo neste quociente.

$$TF_{i,j} = \frac{t_i d_j}{\sum_k t_k d_j}, \tag{2.2}$$

$$IDF_i = log \frac{|D|}{|\{d : t_i\} \in d|}. \tag{2.3}$$

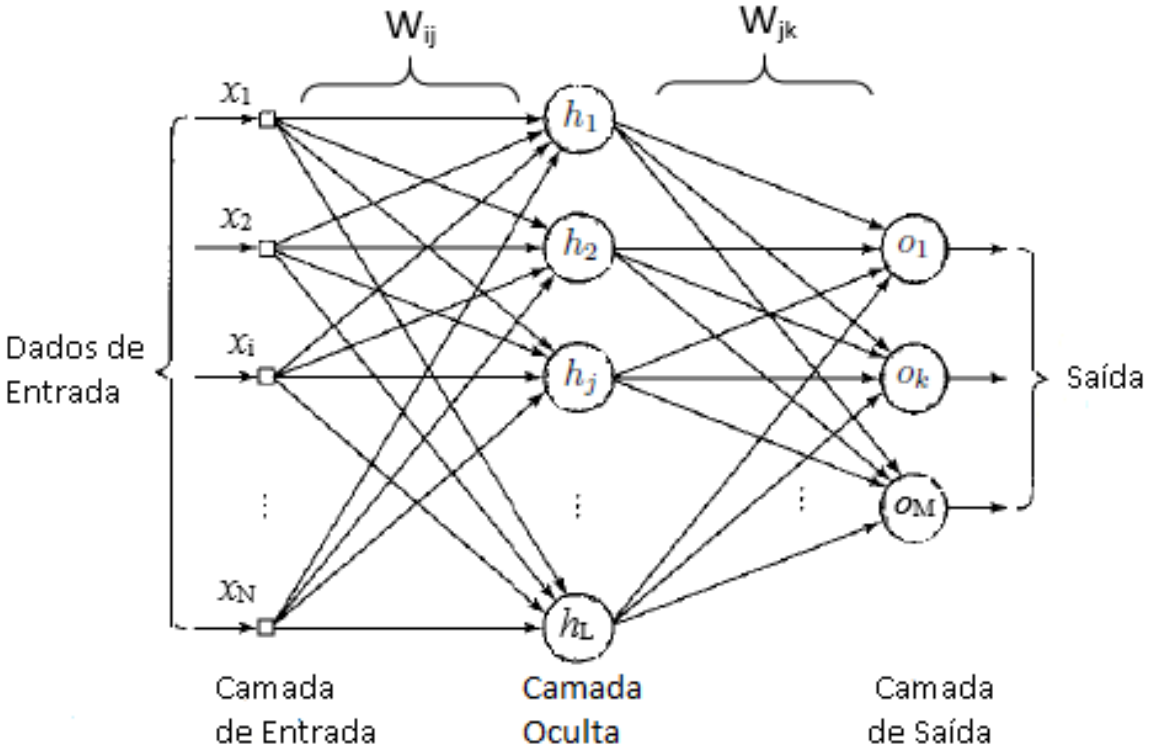
2.3 Redes Neurais Artificiais

As Redes Neurais Artificiais são uma das principais técnicas de Aprendizado de Máquina, e, conforme Carvalho, Braga e Ludermir (2014), são sistemas paralelos e distribuídos compostos por unidades de processamento simples, chamados neurônios, que calculam determinadas funções matemáticas. A maioria dos modelos apresenta conexões associadas a pesos que armazenam o conhecimento representado no modelo e servem para ponderar a entrada recebida da rede.

As RNAs possuem inspiração nas redes neurais biológicas, constituídas de neurônios separados por camadas, que processam informações e estão conectados via pesos sinápticos, sendo na maioria das vezes sistemas adaptativos que modificam sua estrutura através de informações, que fluem pela rede durante a etapa de aprendizado. Este pode ser supervisionado ou não-supervisionado:

- Supervisionado: neste caso as amostras já são rotuladas, e as saídas produzidas pela rede são avaliadas de acordo com as amostras de entrada. A diferença entre o valor desejado

Figura 2: Estrutura de uma Rede Neural MLP



Fonte: Adaptado de Haykin (2001)

(rótulo da amostra) e o valor obtido pela saída da rede é utilizado para atualização dos pesos das conexões;

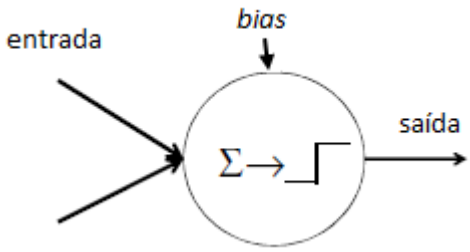
- Não-Supervisionado: as amostras não são rotuladas e a rede necessita por si só identificar relações, padrões ou categorias nos dados apresentados. O objetivo pode ser estimar densidade, extrair características ou redução da dimensionalidade (NOROUZI, 2009).

Existem diversas estruturas de RNAs, sendo as *Multilayer Perceptrons* (MLP) uma das arquiteturas mais utilizadas e que fazem uso do aprendizado supervisionado. As MLPs são amplamente empregadas em modelos de classificação e predição de valores, onde após um processo de treinamento a partir de dados rotulados, espera-se que a rede neural seja capaz de identificar a qual classe as amostras apresentadas na etapa de teste pertencem (ou predizer valores a partir dos dados de entrada).

A estrutura básica de uma MLP é apresentada na Figura 2, na qual é identificada a ocorrência das camadas de entrada, oculta e de saída. Esta particularidade da saída de cada nó servir como entrada para nós da camada seguinte caracteriza a rede como *feed-forward*. Os pesos sinápticos das conexões entre os neurônios ***i*** da camada de entrada aos neurônios ***j*** da oculta estão representados por W_{ij}, já os pesos referentes às conexões dos neurônios da camada oculta com os k neurônios da camada de saída são denotados por W_{jk}.

A camada de entrada é composta por neurônios sensoriais (distribuem o vetor de entrada,

Figura 3: Estrutura de um neurônio artificial



correspondente aos dados de entrada da rede, para os neurônios da camada oculta) e as demais por neurônios computacionais, os quais possuem uma entrada, realizam o processamento e produzem uma saída. A saída da última camada determina a rotulação da amostra ou o valor predito.

A estrutura dos neurônios é exibida na Figura 3, destacando além das já referidas entrada e saída, um valor de *bias*, aplicado para aumentar a entrada líquida da função de ativação (ou diminuir, se este for negativo).

Para o processamento por parte dos neurônios da camada oculta h, cada um destes recebe como entrada o resultado da Equação (2.4)

$$net_j^h = \sum_{i=1}^{\aleph} w_{ij}^h x_i + \theta_j^h, \tag{2.4}$$

onde cada neurônio i da camada de entrada possui sua saída representada por x, os pesos das conexões destes com o neurônio j da camada oculta h representados por w_{ij}^h, e o *bias* associado ao neurônio j representado por θ_j^h.

A este valor é aplicada a função de ativação do neurônio, representada pela Equação (2.5), função esta que pode ser a logística sigmóide, por exemplo, exibida na Equação (2.6).

$$h_j = f_j^h(net_j^h), \tag{2.5}$$

$$sigm = \frac{1}{(1 + e^x)}. \tag{2.6}$$

De forma análoga, o processamento de cada um dos neurônios k da camada de saída o ocorre pela Equação (2.7), com os pesos das conexões entre cada neurônio j da camada oculta com o neurônio k da camada de saída representado por w_{jk}^o, h_j corresponde ao valor de saída do neurônio j da camada oculta e θ_k^o é o *bias* associado ao neurônio da camada de saída. Posteriormente é aplicada a função de ativação, representada pela Equação (2.8), no resultado obtido.

$$net_k^o = \sum_{j=1}^{\aleph} w_{jk}^o h_j + \theta_k^o, \tag{2.7}$$

$$o_k = f^o_k(net^o_k). \tag{2.8}$$

Após este processamento ocorrer por todas as camadas, a última gera uma saída, a qual é comparada com o valor esperado, obtendo então o valor de erro. Este erro é retro-propagado camada por camada a partir da saída (algoritmo *backpropagation*) (HAYKIN, 2001) e empregado para recalcular os pesos, através da regra delta (FREEMAN; SKAPURA, 1991).

O delta utilizado para cálculo da atualização dos pesos que conectam as camadas oculta e de saída se dá pela Equação (2.9), onde η corresponde a uma constante utilizada como taxa de aprendizado, $Y_k - O_k$ é a diferença entre o valor obtido e o desejado e $f^o_k(net^o_k)$ corresponde ao valor de saída dos neurônios da camada de saída.

$$\Delta w_{jk} = -\eta(Y_k - O_k)f^o_k(net^o_k). \tag{2.9}$$

A atualização dos pesos propriamente dita ocorre pela Equação (2.10)

$$w_{jk}(t+1) = w_{jk}(t) + \Delta w_{jk}(t), \tag{2.10}$$

onde os pesos representados w_{jk} no instante t são considerados para os novos valores dos pesos $w_{jk}(t+1)$.

Já para as conexões entre os neurônios das camadas de entrada e oculta se utiliza o cálculo do delta pela Equação (2.11), na qual $\sum_k \delta^o_k w^o_{jk}$ corresponde ao somatório do erro atingido pelas saídas da última camada, e $f^h_j(net^h_j)$ a saída dos neurônios da camada oculta; e a atualização dos pesos ocorre pela Equação (2.12).

$$\delta^h_j = f^h_j(net^h_j) \sum_{k=1}^{M} \delta^o_k w^o_{jk}, \tag{2.11}$$

$$w_{ij}(t+1) = w_{ij}(t) + \eta\delta^h_j x_i. \tag{2.12}$$

Com a regra delta espera-se a diminuição do Erro Quadrático Médio (EQM), exibido na Equação (2.13), o que possibilita o aprendizado.

$$EQM_p = \frac{1}{2}\sum_k (Y_k - O_k)^2. \tag{2.13}$$

Cada repetição do processo de apresentação do conjunto de amostras de treinamento à rede neural e seu ajuste dos pesos é chamado de época. A partir destas repetições ocorre o aprendizado por parte da rede neural, e os critérios de parada podem ser a obtenção de um erro mínimo desejado, um limite máximo de épocas ou um determinado número de épocas sem melhora no resultado. Estes conceitos serviram como base para *Deep Learning*, que emprega a filosofia de RNAs.

2.4 *Deep Learning*

Durante anos foram realizadas, sem êxito, tentativas de métodos de aprendizado em múltiplas camadas (BENGIO, 2009). Foi constatado que a simples adição de camadas a uma rede neural não contribui para um melhor resultado, visto que o método de gradiente descendente (ajuste de pesos buscando minimizar o EQM) tende a ficar preso em mínimos locais ou platôs (ARNOLD et al., 2011), além da considerável elevação do tempo de processamento. Por esta razão, as redes neurais frequentemente são configuradas com uma ou duas camadas ocultas no máximo.

Baseado nos conceitos das RNAs e inspirado em estudos biológicos do cérebro humano e no cortex visual dos mamíferos, surge *Deep Learning*, paradigma que trata a dificuldade que arquiteturas comumente utilizadas, como RNAs ou *Support Vector Machines* (SVMs), por exemplo, possuem em dados com alta dimensionalidade (ARNOLD et al., 2011). Esta é proporcional ao volume do espaço em que os dados estão inseridos, sendo estes mais esparsos quanto maior a dimensionalidade. Do mesmo modo, o algoritmo *backpropagation* apresenta dificuldade nestas arquiteturas com a necessidade de dados rotulados para o treinamento, enquanto que a maior parte dos dados é não-rotulada (HINTON, 2007).

De acordo com Bengio (2009), o foco do aprendizado das arquiteturas que implementam *Deep Learning* é identificar abstrações dos dados, de níveis mais baixos para os mais altos, obtendo novas representações (SOCHER; MANNING, 2013), (MAAS et al., 2011). Deste modo, as características de mais alto nível são formadas pela composição das de mais baixo nível, sendo uma das principais atribuições de *Deep Learning* a extração de características dos dados.

Conforme Deng e Yu (2014), o conceito de *Deep Learning* é abrangente, podendo ser definido como uma sub-área de Aprendizado de Máquina, caracterizado pela:

- utilização de várias camadas de informações não-lineares para extração de características (de forma supervisionada ou não-supervisionada), transformação e análise de padrões;
- uso de algoritmos de aprendizado de múltiplos níveis de representação para identificar relações entre os dados através de modelos estatísticos, normalmente fazendo uso de redes neurais.

Deep Learning teve o seu desenvolvimento favorecido pela busca por algoritmos de aprendizado não-supervisionados para aprender características de dados não-rotulados (YANG, 2013). As estruturas com múltiplas camadas passaram a obter bons resultados após a elaboração do algoritmo de treinamento das *Deep Belief Networks* (DBN), conforme especificado em Hinton, Osindero e Teh (2006). As *Deep Belief Networks* fazem uso de camadas para treinamento não-supervisionado de dados em um momento anterior à classificação, buscando identificar características e uma nova representação dos dados, através das informações contidas nos pesos que interligam as camadas (BENGIO, 2009).

2.4.1 *Deep Belief Networks*

Segundo Lee et al. (2009), as *Deep Belief Networks* são modelos probabilísticos compostos de uma camada visível e várias ocultas. Cada uma destas aprende relações estatísticas entre as camadas mais baixas. Desta forma, as camadas são dispostas hierarquicamente, sendo estas compostas por *Restricted Boltzmann Machines* (RBM). As RBMs (mais detalhadamente abordadas na seção 2.4.1.1) possuem uma camada visível e uma oculta, sendo que somente há conexões (com pesos sinápticos) entre unidades de camadas diferentes.

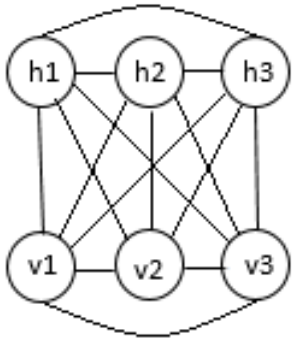
As RBMs são empilhadas, e as unidades ocultas de uma são utilizadas como entrada para outra (a primeira RBM possui os dados de entrada do modelo na camada visível, sendo esta a referida camada visível da DBN). Em Hinton, Osindero e Teh (2006) é proposto um algoritmo guloso de treinamento para DBNs, onde as camadas são treinadas uma a uma na etapa conhecida como *Pre-training*.

Nesta etapa, as RBMs são treinadas individualmente através do algoritmo de *Contrastive Divergence* (CD), Apêndice A, no qual as unidades são processadas e os pesos das conexões entre as unidades visíveis e ocultas atualizados (HINTON, 2002), (HINTON, 2012) e (CARREIRA-PERPINAN; HINTON, 2005). Após o treinamento individual é efetuada a etapa de treinamento supervisionado *Fine-tuning*, onde todos os pesos da estrutura são ajustados.

2.4.1.1 *Restricted Boltzmann Machines*

Restricted Boltzmann Machines são um caso específico de *Boltzmann Machines*, que por sua vez tratam de redes neurais completamente conectadas e recorrentes (em que a saída produzida por um neurônio pode ser retornado para o mesmo neurônio), com unidades visíveis (v) e ocultas (h), conforme ilustrado na Figura 4.

Figura 4: Estrutura de uma *Boltzmann Machine*



As *Boltzmann Machines* também são um caso de *Energy-Based Models* (EBM). Segundo Bengio (2009), EBMs são modelos que associam um escalar de energia para cada conjunto de variáveis (no caso de RBM, unidades visíveis e ocultas). Quando o EBM é probabilístico é definida a distribuição de probabilidade através de uma função de energia. No caso de uma

Boltzmann Machine, esta se dá pela Equação (2.14).

$$E(v,h) = -b^{'}v - c^{'}h - h^{'}Wv - v^{'}Uv - h^{'}Vh. \tag{2.14}$$

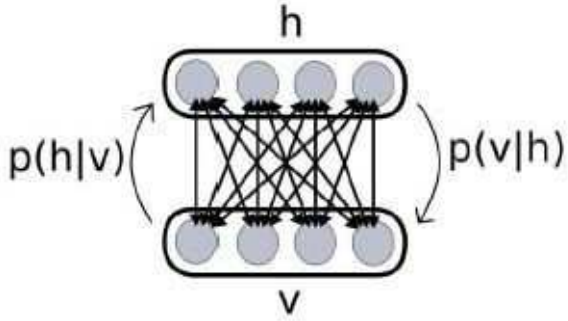
Nesta equação, v e h são os vetores representando as unidades visíveis e ocultas, respectivamente ($v^{'}$ e $h^{'}$ suas respectivas matrizes transpostas); $b^{'}$ e $c^{'}$ são matrizes transpostas dos vetores de *bias* associados a cada elemento dos vetores v e h; W é a matriz de pesos das conexões entre as camadas visíveis e ocultas e as matrizes U e V representam os pesos das conexões entre unidades da mesma camada, U nas visíveis e V para as ocultas.

A estrutura de uma RBM é composta de um grafo direcionado bipartido (Figura 5), sendo uma rede neural com somente uma camada oculta, que tenta encontrar semelhanças nos dados de entrada de maneira não-supervisionada. A restrição que dá nome às RBMs consiste em não serem permitidas conexões entre neurônios da mesma camada. Com isso, a função de energia da RBM é diferenciada, exibida na Equação (2.15)

$$E(v,h) = -\sum_{i} b_i v_i - \sum_{j} c_j h_j - \sum_{i,j} w_{ij} v_i h_j, \tag{2.15}$$

onde v e h são as unidades visíveis e ocultas, respectivamente; i e j referem-se aos índices das unidades; w_{ij} corresponde ao peso da conexão entre v_i e h_j; e b_i e c_j são *bias* associados às unidades v_i e h_j, respectivamente. Nesta equação é verificada a ausência dos termos $v^{'}Uv$ e $h^{'}Vh$, correspondentes às conexões entre unidades da mesma camada, presentes nas *Boltzmann Machines* e não existentes nas RBMs.

Figura 5: Estrutura de uma RBM



Fonte: Adaptado de Arnold (2013)

Com o valor de energia é possível calcular a distribuição p para o vetor v, conforme Equação (2.16)

$$p(v) = \frac{\sum_{h} e^{-E(v,h)}}{\sum_{u,g} e^{-E(u,g)}}, \tag{2.16}$$

onde v e h são os já mencionados valores das unidades visíveis e ocultas e u e g referem-se aos conjuntos que englobam todas as unidades visíveis e ocultas, respectivamente.

Já a computação dos valores para as unidades $p(v|h)$ e $p(h|v)$, observados na Figura 5,

ocorre através das Equações (2.17) e (2.18) para as unidades visíves e (2.19) e (2.20) para as ocultas, onde para computar h são considerados todos os índices i do vetor v e de forma análoga, todos os índices j do vetor h para calcular os valores de v.

$$p(v|h) = \prod_i p(v_i|h), \tag{2.17}$$

$$p(v_i = 1|h) = sigm(a_j + \sum_j h_j w_{ij}), \tag{2.18}$$

$$p(h|v) = \prod_j p(h_j|v), \tag{2.19}$$

$$p(h_j = 1|v) = sigm(b_j + \sum_i v_i w_{ij}). \tag{2.20}$$

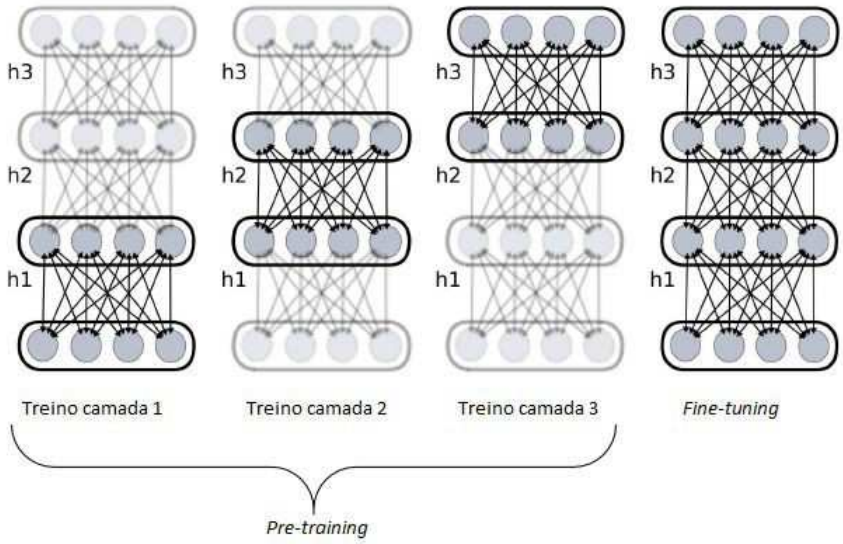
Na DBN, após cada RBM ser treinada, sua camada com unidades ocultas serve de entrada para a RBM que compõe a próxima camada da DBN, que por sua vez terá seu treinamento e posteriormente suas unidades ocultas alimentando uma próxima camada de RBM. Esta situação pode se repetir sucessivamente e por isso diz-se que as RBMs encontram-se empilhadas.

Este processo no qual as camadas são treinadas individualmente faz parte da já referida etapa de *Pre-training*, destacando-se como uma das vantagens do treinamento individual das camadas a melhor inicialização dos pesos que as conectam (BENGIO et al., 2007). Após este processo, tem-se os neurônios que anteriormente compunham as RBMs conectados camada após camada por conexões com pesos sinápticos. A partir disso é efetuado o processo de *Fine-tuning*, ajustando os pesos de toda a estrutura via algoritmo *backpropagation* (não mais considerando RBMs individualmente) com o método do gradiente conjugado (GONG; SHAO; XU, 2010). As etapas de *Pre-training* e *Fine-tuning* são ilustradas na Figura 6.

Este modelo foi adotado em diferentes áreas, como reconhecimento de dígitos manuscritos para o conjunto MNIST [1] (amplamente utilizados em abordagens de *Deep Learning*) (HINTON; SALAKHUTDINOV, 2006) e em Salakhutdinov e Hinton (2009a), onde a primeira camada foi treinada por um modelo de *Poisson* para formar um vetor de palavras com suas quantidades de ocorrências, para posterior agrupamento de documentos por similaridade. Na literatura corrente identifica-se variações inclusive das unidades que compõem estas estruturas de *Deep Learning*, destacando, a título de ilustração, o trabalho de Mnih, Larochelle e Hinton (2012) que abordou uma variante das RBMs: a *Conditional Restricted Boltzmann Machines* (CRBM), exibida na Figura 7, na qual é observada a presença da estrutura u, utilizada para determinar dinamicamente os *bias* ou pesos da RBM.

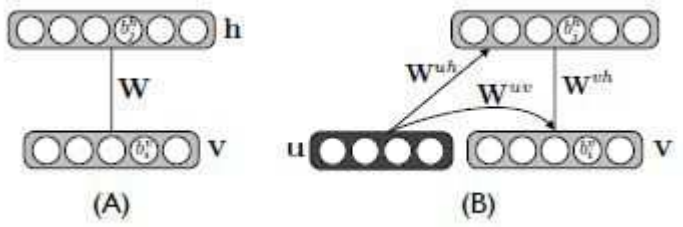
[1] http://yann.lecun.com/exdb/mnist/

Figura 6: *Pre-training* e *Fine-tuning*



Fonte: Adaptado de Arnold (2013)

Figura 7: Diferença entre RBM (A) e CRBM (B)



Fonte: Mnih, Larochelle e Hinton (2012)

2.5 Métricas para Análise de Resultados

Realizadas as etapas de pré-processamento e Mineração de Dados, independente da aplicação de *Deep Learning*, o modelo classificador produzirá uma saída, utilizada para indicar a classe a qual cada amostra (opinião) pertence. É necessária a análise dos valores desta saída para avaliar o comportamento do classificador.

Os resultados obtidos são comparados com os valores desejados e obtém-se então a quantidade de amostras classificadas corretamente e erroneamente, para cada classe avaliada. Existem métricas capazes de avaliar o comportamento do classificador (LIU, 2012), sendo que estas podem ser calculadas pelos valores de VP, VN, FP e FN, descritos abaixo.

No caso de polaridade, amostras positivas classificadas corretamente compõem o valor de Verdadeiras Positivas (VP) e as negativas erroneamente preditas como positivas o de Falsas Positivas (FP). De forma análoga, as amostras classificadas corretamente como negativas fazem parte do valor de Verdadeiras Negativas (VN) e as positivas preditas como negativas compõem

o de Falsas Negativas (FN). Os valores VP, FP, VN e FN são utilizados para montar a matriz de confusão, conforme Tabela 1.

Tabela 1: Matriz de Confusão

Resultado desejado	Resultado predito	
	Positivo	Negativo
Positivo	VP	FN
Negativo	FP	VN

Através dos valores componentes da matriz podem ser calculadas diferentes métricas, destacando as utilizadas neste trabalho:

- acurácia: efetua uma relação entre amostras classificadas corretamente e o total de amostras;
- precisão: determina o quanto o classificador é capaz de rejeitar as outras classes (em relação à classe a qual a precisão foi calculada);
- *recall*: capacidade de classificação de amostras pertencentes à classe analisada;
- *f-measure*: também chamada de *f-score*, é a média harmônica das medidas de precisão e *recall*.

Ao contrário dos trabalhos correlatos (apresentados no Capítulo 3), em que se utilizou apenas a acurácia como unidade de medida, foram computadas as demais unidades mencionadas, visto que quando apenas a acurácia é abordada o comportamento do classificador pode não ser precisamente avaliado, principalmente com dados desbalanceados (com quantidades diferentes de amostras entre as classes). Esta situação pode ser facilmente identificada em um cenário, por exemplo, em que 90% das amostras do conjunto de teste pertence à classe *positiva* e 10% à *negativa*: se ao efetuar a avaliação, 100% das amostras forem classificadas como positivas, a acurácia calculada será de 90%. Apesar deste percentual alto, o classificador pode não estar apto a identificar opiniões negativas.

As fórmulas de acurácia, *recall*, precisão e *f-measure* são expostas nas Equações de (2.21) a (2.28), sendo que *recall* e precisão podem ser avaliadas por classe, pos e neg para *positiva* e *negativa*, respectivamente (READ et al., 2009).

$$acuracia = \frac{VP + VN}{VP + FP + VN + FN}, \tag{2.21}$$

$$recall_{pos} = \frac{VP}{VP + FN}, \tag{2.22}$$

$$recall_{neg} = \frac{VN}{VN + FP}, \tag{2.23}$$

$$precisao_{pos} = \frac{VP}{VP + FP}, \tag{2.24}$$

$$precisao_{neg} = \frac{VN}{VN + FN}, \tag{2.25}$$

$$recall = \frac{recall_{pos} + recall_{neg}}{2}, \tag{2.26}$$

$$precisao = \frac{precisao_{pos} + precisao_{neg}}{2}, \tag{2.27}$$

$$fmeasure = \frac{2 * recall * precisao}{recall + precisao}. \tag{2.28}$$

2.6 Considerações

Neste capítulo foram abordados conceitos necessários para o melhor entendimento da pesquisa realizada neste trabalho. Inicialmente foi apresentada a grande área de KDD para posteriormente contextualizar Mineração de Opiniões. Foram apresentados também problemas inerentes à Mineração Textual, juntamente com procedimentos da etapa de pré-processamento, destacando-se a seleção de termos através do IG. Estes procedimentos aplicados aos dados utilizados efetuaram o refinamento o qual será avaliado em relação à utilização apenas de *Deep Learning* na identificação de características dos dados.

Exploraram-se também conceitos de Redes Neurais Artificiais, *Deep Learning*, sua implementação pelas *Deep Belief Networks* e as unidades que as compõem (*Restricted Boltzmann Machines*). Foram apresentadas as etapas de treinamento, destacando-se a *Pre-training* que atua em uma das principais atribuições de *Deep Learning*: extração de características de dados. Esta atribuição é um dos pontos enfatizados no presente trabalho, em que a extração de características ocorre em dados já refinados.

Por fim foram mencionadas métricas utilizadas para avaliação dos classificadores. Estes conceitos são abordados nos trabalhos relacionados apresentados no Capítulo 3 e são base para a metodologia aplicada no presente trabalho, indo ao encontro da implementação utilizada para os experimentos descritos no Capítulo 4.

3 TRABALHOS RELACIONADOS

Neste capítulo são apresentados três trabalhos que propuseram modelos que implementam *Deep Learning*, dos quais dois tratam da identificação de polaridade: em Zhou, Chen e Wang (2010) foi proposto o modelo de *Active Learning* para classificação de opiniões e Glorot, Bordes e Bengio (2011a) criaram um modelo de *Deep Learning* no qual testaram-se ativações de neurônios com diferentes funções e se levantou a questão do quanto a etapa de *Pre-training* influencia nos resultados.

Também é demonstrado o trabalho de Glorot, Bordes e Bengio (2011b), onde foi implementado um modelo para classificação de opiniões referentes a diferentes domínios. Estes trabalhos comprovam que *Deep Learning* pode ser aplicado também em opiniões, com os seus resultados, juntamente com os obtidos através de experimentos realizados com a ferramenta AlchemyAPI[1], servindo como *benchmark* para este trabalho.

3.1 *Active Deep Network*

Zhou, Chen e Wang (2010) propuseram um modelo semi-supervisionado de aprendizado, chamado de *Active Deep Network* (ADN), para classificar opiniões. Os experimentos foram realizados para os dados de filmes de Pang e Lee (2004) e quatro bases de dados referentes a diferentes tipos de produtos: livros (BOO), DVDs (DVD), eletrônicos (ELE) e utensílios de cozinha (KIT), com cada conjunto composto por 2000 opiniões igualmente divididas em positivas e negativas.

A etapa de pré-processamento foi semelhante à utilizada por Dasgupta e Ng (2009), onde as opiniões são representadas por vetores binários representando a presença e a ausência dos termos. Os termos foram selecionados por frequência, descartando os 1,5% mais frequentes (por presumir que estes tratam-se de *stopwords*).

O treinamento é dito semi-supervisionado, pois apenas parte de amostras são rotuladas para calcular os pesos. Na etapa de *Pre-training* (apresentada no Capítulo 2), todas as amostras participam do treinamento das camadas da arquitetura de *Deep Learning* para gerar a matriz de pesos *W*.

Através da técnica de *Active Learning*, implementada no trabalho, são escolhidas as amostras dentre os conjuntos de treinamento a serem consideradas com o seu rótulo para o treinamento. São selecionadas as amostras possivelmente mais difíceis de classificar, com esta dificuldade consistindo na medida da distância da amostra para o separador de classe no hiperplano (quanto mais próxima do separador, maior a dificuldade de classificá-la). Então o modelo passa por um treinamento considerando as amostras selecionadas pelo algoritmo de *Active Learning*.

Os experimentos foram realizados dividindo-se randomicamente os conjuntos em 10-*folds*, para posterior teste via *cross-validation*. Os resultados foram comparados com outros modelos

[1] http://www.alchemyapi.com/

referenciados, destacados na Tabela 2 apenas os resultados do modelo ADN e de DBN para as opiniões de filmes, utensílios de cozinha, eletrônicos, livros e DVDs.

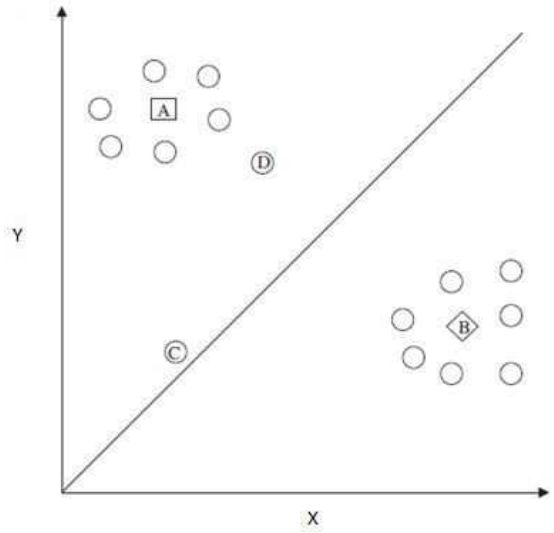
Tabela 2: Acurácia (%) obtida pelo modelo de *Active Deep Network* e DBN

	MOV	KIT	ELE	BOO	DVD
DBN	71,3	72,6	73,6	64,3	66,7
ADN	76,3	77,5	76,8	69,0	71,6

Fonte: Zhou, Chen e Wang (2010)

Uma extensão do trabalho acima apresentado foi proposto em Zhou, Chen e Wang (2013), a *Information Active Deep Network* (IADN), a qual utiliza a densidade da informação das amostras não-rotuladas. O conceito de densidade de informação pode ser visualizado na Figura 8, em que observa-se as amostras *A*, *C* e *D* pertencentes a uma classe e a amostra *B* a outra. Pelo algoritmo ADN, a amostra *C* seria selecionada para passar pelo aprendizado supervisionado, por estar mais próxima do separador de classes. No entanto, *D* está mais próxima do centro de sua classe, possuindo mais informação sobre o conjunto de dados, o que ocasionaria sua seleção pelo algoritmo IADN, pois este não considera apenas a distância para o separador.

Figura 8: Exemplo de amostras para ilustrar densidade de informações



Fonte: Zhou, Chen e Wang (2013)

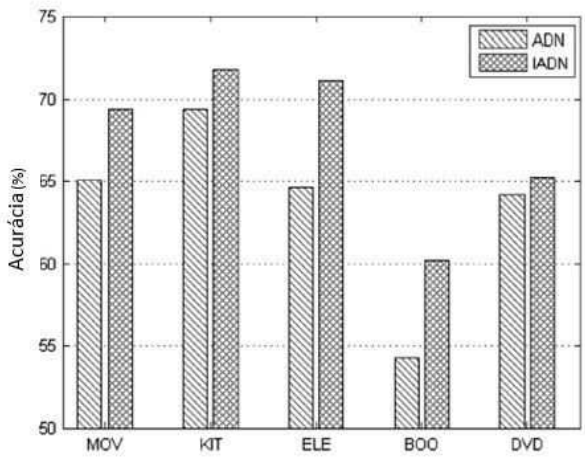
Quando utilizadas 100 amostras rotuladas para os cinco conjuntos de dados, obteve-se os resultados exibidos na Tabela 3, superando os verificados na Tabela 2. Os resultados de IADN possuem melhora expressiva em relação à ADN quando selecionadas 10 amostras rotuladas, sendo exibido o gráfico comparativo na Figura 9, em que a acurácia obtida pelo modelo que utilizou IADN foi significativamente maior em todos os casos, exceto para a base de dados de DVD, com os resultados mais próximos.

Tabela 3: Acurácia (%) obtida pelo modelo IADN, com 100 amostras rotuladas

	MOV	KIT	ELE	BOO	DVD
IADN	76,4	78,2	77,9	69,7	72,2

Fonte: Zhou, Chen e Wang (2013)

Figura 9: Comparação de resultados ADN x IADN, com 10 amostras rotuladas



Fonte: Zhou, Chen e Wang (2013)

3.2 Rede Neural Retificadora

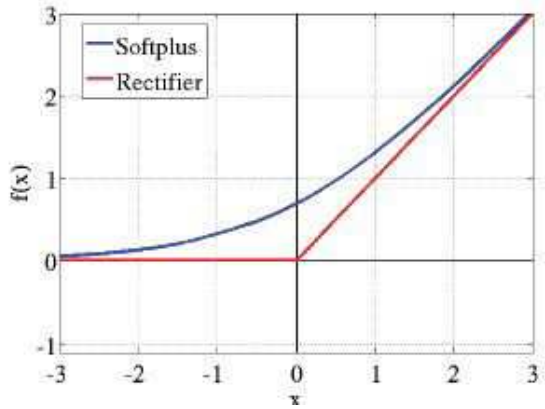
Glorot, Bordes e Bengio (2011a) apresentaram a utilização da função *rectifier* para ativação dos neurônios, analisando também a possibilidade de se atingir bons resultados em uma estrutura de *Deep Learning*, sem a etapa de *Pre-training*. O trabalho é uma extensão de Nair e Hinton (2010).

As redes retificadoras são uma alternativa para modelos de neurônios biológicos. A função retificadora permite uma representação esparsa dos dados, sendo esta uma inspiração biológica, visto que estudos sobre o gasto energético do cérebro têm apontado que os neurônios armazenam as informações de forma esparsa (ATTWELL; LAUGHLIN, 2001).

Uma das vantagens desta representação é a possibilidade de manter as variações presentes nos dados, visto que quando estes são muito densos, pequenas mudanças podem impactar na representação dos dados por vetores. Além disso, conforme os autores do trabalho, dados esparsos possuem maior propensão a serem linearmente separáveis, pelo fato dos dados estarem representados em um espaço multi-dimensional.

Buscando atenuar os efeitos da ativação da função *rectifier*, estudaram uma forma mais suave desta, a *softplus*, sendo ambas expressas nas Equações (3.1) e (3.2), com seus respectivos gráficos exibidos na Figura 10. Analisando os gráficos, percebe-se a possibilidade de tratamento de valores negativos na *softplus*, enquanto que a *rectifier* resulta em *0* sempre que *x* é negativo.

Figura 10: Gráfico das funções *rectifier* e *softplus*



Fonte: Glorot, Bordes e Bengio (2011a)

$$rectifier(x) = max(0, x), \tag{3.1}$$

$$softplus(x) = log(1 + e^x). \tag{3.2}$$

Os resultados obtidos com estas funções foram comparados com os atingidos pela tangente hiperbólica (Tanh), função mais comumente empregada, exibida na Equação (3.3).

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \tag{3.3}$$

Foram utilizadas quatro bases de dados: MNIST, imagens de dígitos; CIFAR10, imagens RGB; NISTP imagens de caracteres e NORB, figuras de brinquedos. Os valores da raiz do EQM são exibidos na Tabela 4, comparando a utilização das etapas do processo de *Deep Learning* e sem a etapa de *Pre-training*, verificando que na presença desta etapa o erro foi menor na maioria dos casos.

Tabela 4: Erro de teste (%) na aplicação das diferentes funções

Com *Pre-training*				
Neuron	MNIST	CIFAR10	NISTP	NORB
Rectifier	1,2	49,96	32,86	16,46
Tanh	1,16	50,79	35,89	17,66
Softplus	1,17	49,52	33,27	19,19
Sem *Pre-training*				
Neuron	MNIST	CIFAR10	NISTP	NORB
Rectifier	1,43	50,86	32,64	16,40
Tanh	1,57	52,62	36,46	19,29
Softplus	1,77	53,20	35,48	17,68

Fonte: Glorot, Bordes e Bengio (2011a)

Já analisando as diferentes funções de ativação, identifica-se que na ausência da etapa de

Pre-training a função *rectifier* obteve os melhores resultados em todos os casos. Entretanto, na presença da etapa de *Pre-training* isto repetiu-se apenas para os conjuntos NISTP e NORB.

Além dos conjuntos de imagens, aplicou-se o modelo na classificação de sentimentos em dados obtidos de restaurantes[2] (10 mil opiniões rotuladas e 300 mil não-rotuladas) e nos dados de filmes, em uma abordagem semelhante a utilizada em Zhou, Chen e Wang (2010), atingindo nestes a acurácia de 78,95%.

3.3 Adaptação de Domínio

Quando se trata de polaridade de opiniões pode-se encontrar semelhanças entre conjuntos de opiniões de diferentes domínios. Buscando esta identificação tem-se utilizado *Deep Learning*, dado o sucesso na representação de dados e identificação de similaridade entre estes.

Em Glorot, Bordes e Bengio (2011b) foi proposto um modelo de *Deep Learning* para classificação de opiniões de diferentes domínios: equipamentos de cozinha, eletrônicos, DVDs, livros, entre outros, todos pertencentes à base de dados da *Amazon*.

Na etapa de pré-processamento, as opiniões foram tratadas como *bag-of-words* e transformadas em vetores binários que representam ausência ou presença dos termos (unigramas ou bigramas), sendo selecionados os 5000 termos mais frequentes.

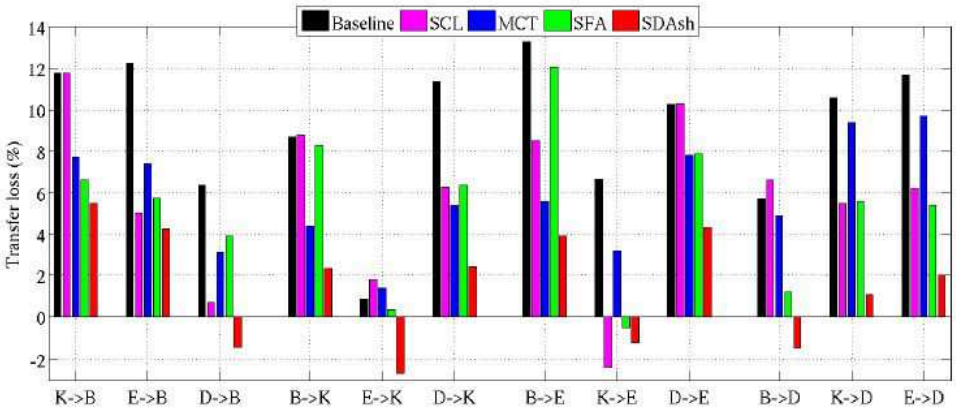
O modelo proposto fez uso de *Stacked Denoised Auto-encoder* (SDA) (VINCENT et al., 2008), estruturas que possibilitam a redução da dimensionalidade de dados e favorecem procedimentos como visualização e armazenamento de dados multi-dimensionais, semelhantes às DBNs, com as camadas compostas por *Auto-encoders* em vez de RBMs. De acordo com Arnold (2013), os *Auto-encoders* são redes neurais que aprendem representações dos dados minimizando o erro de reconstrução destes, enquanto que as *Boltzmann Machines* são redes neurais probabilísticas que definem a distribuição da entrada no espaço, sendo as RBMs aproximadores universais. A ideia do *Auto-encoder* foi obtida através da busca de uma generalização não linear de *Principal Component Analysis* (PCA) (JOLLIFFE, 1986), ferramenta que possibilita representar de forma reduzida um grande volume de dados, enquanto que as RBMs, segundo Vico (2012), são mais próximas das *Independent Component Analysis* (ICA).

Os *Auto-encoders* são compostos pelas funções *h(.)* (*encoder*) e *g(.)* (*decoder*), para reduzir a dimensionalidade dos dados e reconstruir os dados originais, respectivamente. A reconstrução da entrada *x* é dada por $r(x) = g(h(x))$ e os *Auto-encoders* são então treinados para diminuir o erro de reconstrução $loss(x, r(x))$.

O algoritmo de treinamento é semelhante ao dos trabalhos relacionados já citados, com as unidades treinadas individualmente de forma não-supervisionada na etapa de *Pre-training*. Destaca-se ainda que a função utilizada para o *decoder* foi a versão suave da *rectifier* (exceto na primeira camada, na qual optou-se pela logística sigmóide), tratada no trabalho relacionado já apresentado.

[2]www.opentable.com

Figura 11: *Transfer loss* para os diferentes domínios, comparando diferentes técnicas



Fonte: Glorot, Bordes e Bengio (2011b)

Para a adaptação do domínio propriamente dita utilizou-se um domínio (utensílios de cozinha, eletrônicos, DVDs ou livros) como *Source S*, com dados rotulados que passaram por uma etapa de treinamento pelo classificador linear SVM. Os demais conjuntos *Target T* de dados passaram pela estrutura SDA para posteriormente serem testadas no SVM já treinado por *S*. Esse procedimento se repetiu para todos os domínios.

Entre as unidades de medida utilizadas para verificar o erro de reconstrução dos dados de entrada, destaca-se *transfer loss*, que é a diferença entre erro de teste após o conjunto *Target* ser treinado no modelo de *Deep Learning* e testado no modelo gerado pelo SVM; e o erro de teste após *Target* ser treinado e testado no SVM. Esta medida pode possuir valor negativo, quando o erro de reconstrução de treino e teste de *Source* e *Target* é menor que quando *Target* é treinado e testado no SVM.

Os resultados foram comparados com demais modelos que abordaram adaptação de domínio, como *Multi-label Consensus Training* (MCT) (LI; ZONG, 2008), *Structural Correspondence Learning* (SCL) (BLITZER; DREDZE; PEREIRA, 2007) e *Spectral Feature Alignment* (SFA) (PAN et al., 2010), sendo exibido o gráfico com os resultados de *transfer loss* para as bases de cozinha (K), eletrônicos (E), DVDs (D) e livros (B) na Figura 11.

Foram realizados experimentos em que os domínios alternaram entre *Source* e *Target*, e na maioria dos casos, o modelo SDAsh (SDA proposto no modelo) obteve o melhor resultado, possuindo valor negativo por vezes, indicando que o erro de teste após o treinamento do modelo proposto foi menor que o conjunto sendo treinado e testado pelo SVM.

3.4 Considerações

Neste capítulo foram apresentados três trabalhos que aplicaram *Deep Learning* em Mineração de Opiniões. Abordaram-se diferentes implementações dos conceitos de *Deep Learning*, visto que foram utilizados *Stacked Auto-encoders* e *Deep Belief Networks*.

Os trabalhos relacionados ainda estimulam a experimentação com variações nos modelos

que aplicam *Deep Learning*, como em Glorot, Bordes e Bengio (2011a), que analisaram diferentes funções de ativação e alternaram experimentações com a presença e ausência da etapa de *Pre-training*. Como esta etapa é uma característica do algoritmo de treinamento das *Deep Belief Networks*, juntamente com os melhores resultados obtidos com a sua utilização no trabalho relacionado, na modelagem realizada se fez uso dela. Destaca-se também a possibilidade de combinação de técnicas, como em Zhou, Chen e Wang (2010), que utilizaram *Deep Learning* em conjunto com um método de seleção de amostras para o treinamento supervisionado (*Active Learning*).

Os trabalhos mencionados, no entanto, não demonstram a obtenção de resultados próximos aos encontrados na literatura corrente, se tratando de classificação de opiniões, que em geral atingem melhores resultados (HE; LIN; ALANI, 2011), (PALTOGLOU; THELWALL, 2010) e (YESSENALINA; YUE; CARDIE, 2010). Enquanto que no trabalho de Zhou, Chen e Wang (2010) é enfatizado o algoritmo de *Active Learning*, e em Glorot, Bordes e Bengio (2011a) a Mineração de Opiniões é abordada de forma secundária (pois a pesquisa trata também de *Deep Learning* aplicada a reconhecimento de imagens e dígitos), os valores de acurácia atingem no máximo 76,4% para a base de dados clássica de filmes de Pang e Lee (2004).

Além destes trabalhos terem obtido resultados inferiores a outras aplicações de Mineração de Opiniões, estes resultados são relatados de forma muito sucinta, com poucas variações nas configurações para os experimentos efetuados e com pouco detalhamento. Também destaca-se como ponto negativo que a única unidade de medida apresentada foi a acurácia.

No presente trabalho são exploradas diferentes configurações para a implementação da DBN de Ruslan Salakhutdinov e Geoffrey Hinton, também adaptada para experimentação no trabalho de Zhou, Chen e Wang (2010), além de apresentar outras unidades de medidas, separando-as por classe inclusive (como *recall* e precisão), fornecendo maiores subsídios para a análise da efetividade dos classificadores. Além disso, o presente trabalho procura abordar de forma mais rígida a aplicação de *Deep Learning* em Mineração de Opiniões, buscando possibilidades de melhorar os resultados até então obtidos via variabilidade na parametrização.

Neste sentido, já foi relatado que *Deep Learning* se destaca por sua capacidade de extração de características dos dados, sendo que nos trabalhos relacionados foi realizado um pré-processamento básico dos dados, considerando no máximo a frequência para seleção de termos, deixando a cargo de *Deep Learning* a identificação das características dos dados, o que também ocorreu no presente trabalho, mas aplicando-a em dados já refinados.

4 MODELAGEM PROPOSTA E RESULTADOS OBTIDOS

Neste capítulo é apresentada a metodologia utilizada no presente trabalho, desde a etapa de pré-processamento até a forma de análise dos resultados. Também são relatados os experimentos computacionais juntamente com os resultados obtidos.

4.1 Metodologia

A abordagem proposta engloba o emprego de *Deep Learning* buscando verificar a viabilidade de sua aplicação no processo de classificação de opiniões fazendo uso de dados refinados, ao contrário da literatura corrente (onde a seleção de características é feita pela técnica de *Deep Learning* em dados brutos), no sentido de realizar uma análise crítica comparando as abordagens.

Os dados considerados nas experimentações são de domínio público, encontrando-se disponíveis para download, amplamente utilizados em aplicações de Mineração de Opiniões: base de dados de opiniões referentes a filmes e outras relacionadas a diferentes produtos (GPS, livros e câmeras), das quais a primeira é oriunda de Pang e Lee (2004) e as demais provenientes do *e-commerce Amazon*.

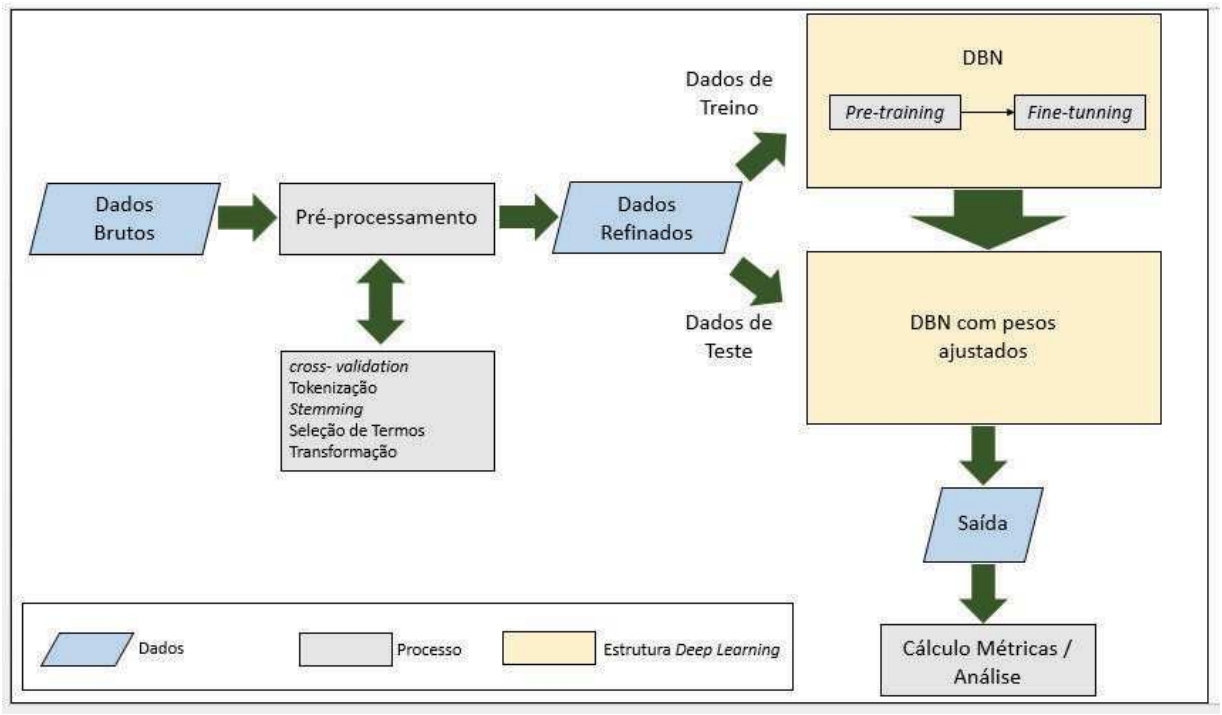
O conjunto de Pang e Lee (2004) é composto por 2000 opiniões igualmente divididas em positivas e negativas. Os pertencentes à base de dados do *e-commerce* são compostos pelo mesmo número de amostras, entretanto estas são classificadas com rótulos variando de uma a cinco estrelas, da mais negativa para a mais positiva, respectivamente. Desta forma, opiniões com uma e duas estrelas foram agrupadas em negativas enquanto as que possuem quatro ou cinco em positivas. As opiniões com três estrelas foram descartadas.

Na Figura 12 é exibida a estrutura da modelagem, onde verifica-se que os dados passaram pelo método *cross-validation*, compondo os 10-*folds*, e pelos procedimentos mencionados no Capítulo 2, como tokenização e *stemming*, na etapa de pré-processamento. Nesta etapa ainda realizou-se a seleção de termos, na qual se empregou o *Information Gain*, criando diferentes conjuntos de dados, cada um correspondendo a uma quantidade de termos. Os dados ainda passaram pelo processo de transformação, no qual dividiram-se treino e teste com os valores de TF-IDF, TF e frequência de cada termo para todas as opiniões. Estes passos de pré-processamento, seleção de termos e transformação foram efetuados no trabalho de Moraes (2013).

Com os dados já refinados, os conjuntos de treino e teste serviram de entrada para uma arquitetura *Deep Learning*, com n camadas que passam por treinamento camada após camada e ajustes de pesos pelo algoritmo *backpropagation*. Respeitando os modelos analisados da literatura até então, inclusive os trabalhos relacionados no Capítulo 3, foram adotadas três camadas ocultas para os experimentos. Tal medida possibilita uma análise mais justa das alterações de parametrizações da estrutura do modelo, como épocas e número de neurônios por camada.

A primeira camada da estrutura é uma RBM, na qual os dados de entrada compõem as uni-

Figura 12: Estrutura da modelagem



dades visíveis. Estes valores, juntamente com os pesos sinápticos, são computados e servem de entrada para unidades ocultas da primeira RBM, que por sua vez também realizam o processa- mento, sendo um processo iterativo, repetido tantas vezes de acordo com o número de épocas parametrizado.

Ao fim deste processo, as unidades ocultas da primeira RBM são utilizadas como entrada para a segunda, possuindo esta suas respectivas unidades ocultas que posteriormente alimentam a terceira. A quantidade de unidades visíveis da primeira RBM varia de acordo com o número de termos do conjunto de dados utilizado, enquanto que a das demais RBMs é parametrizada pela aplicação desenvolvida.

Após a inicialização dos pesos pela etapa de *Pre-training*, é realizada a etapa de *Fine-tuning*, gerando um modelo com os pesos ajustados (conforme Figura 12). Assim os dados percorrem a estrutura produzindo uma saída, posteriormente analisada para verificar o comportamento do classificador.

Com a aplicação do método *k-fold cross validation*, o conjunto de treino é dividido em *k* partes iguais e o treinamento realizado *k* vezes, onde em cada vez é utilizada de *k* - 1 partes para o treino e a parte restante para teste, tendo ao final *k* conjuntos de treino e teste. Esta etapa é realizada para evitar que no conjunto de teste ocorram características presentes no de treino, mantendo-os independentes. Assim, para os experimentos realizados as métricas mencionadas são valores médios obtidos para os *k* conjuntos.

O processo de treino e teste ocorre separadamente para cada conjunto com *n* termos. Para este há um laço iterado a cada *fold*, onde ao término da iteração são armazenados os valores das medidas (descritas na seção abaixo) para posteriormente calcular as médias do conjunto todo.

4.1.1 Análise dos Resultados

No final de cada iteração são comparados os valores da saída da estrutura com os desejados (as opiniões de teste possuem um conjunto correspondente indicando se cada opinião é positiva ou negativa), para verificar se o modelo classificou corretamente a opinião. Nesta etapa aplica-se o conceito da Matriz de Confusão apresentada no Capítulo 2. Através destes valores calcula-se as medidas de acurácia (única apontada nos trabalhos relacionados), precisão, *recall* e *f-measure*, possibilitando assim a comparação dos resultados para analisar a eficiência da aplicação de *Deep Learning* em dados refinados. Destaca-se também a análise das variações das quantidades de termos e de neurônios por camada oculta, buscando identificar a melhor configuração.

4.2 Experimentos Computacionais

Nas seções seguintes são apresentados os dados utilizados, as ferramentas aplicadas, juntamente com os experimentos realizados, seguidos pelos resultados atingidos, com uma análise crítica dos mesmos.

4.2.1 Dados Utilizados

Conforme já mencionado, a etapa de pré-processamento ocorreu no trabalho de Moraes (2013), onde os dados foram inicialmente transformados para o formato do Matlab[1], através da ferramenta TMG[2], já aplicando técnicas de pré-processamento. Posteriormente foi realizada a etapa de cálculo de IG para a seleção de termos, criando conjuntos com diferentes quantidades de termos (30, 50, 100, 200, 300, 400, 500, 1000, 2000, 3000, 5000 e 20000). Além desta distinção, os dados foram separados em 10-*folds*, com cada um destes possuindo conjuntos de treino e teste. Estes são compostos por matrizes de *n* x 1800 (treino) e *n* x 200 (teste), onde 1800 e 200 correspondem às quantidades de amostras (opiniões) e *n* às de termos. Para as experimentações foram utilizados os conjuntos de treino e teste com os valores de TF, devido à obtenção de melhores resultados em experimentos preliminares (em relação à frequência e TF-IDF).

Em Moraes (2013) também foram gerados conjuntos de dados em que o número de amostras positivas e negativas é diferente, ou seja, dados desbalanceados. Tal medida foi adotada para simular diferentes cenários de desbalanceamento e de tratamento deste por *undersampling*, técnica na qual se retira amostras da classe majoritária, buscando tornar mais próximas as quantidades de amostras positivas e negativas.

No entanto, para o presente trabalho somente é abordado o contexto de dados balanceados,

[1] www.mathworks.com

[2] http://scgroup20.ceid.upatras.gr:8000/tmg/

mantendo a possibilidade dos cenários de desbalanceamento e *undersampling* serem adotados em experimentos futuros.

4.2.2 Ferramentas

Diferentes ferramentas que implementam algoritmos de *Deep Learning* encontram-se disponíveis e foram avaliadas, como *Theano* [3] e *DeepLearnToolbox* [4]. A primeira não é totalmente portável, visto que possui limitações para o sistema operacional *Microsoft Windows*, além de requerer a plataforma *Python*[5] de desenvolvimento. Já a segunda é um pacote de funcionalidades para o Matlab que inclui implementações de *Convolutional Auto-encoders* (CAE), *Convolutional Neural Networks* (CNN), *Deep Belief Networks* (DBN), *Neural Networks* (NN) e *Stacked Auto-encoders* (SAE), as quais foram desenvolvidas utilizando a base de dados de dígitos manuscritos MNIST.

Após instalada a referida biblioteca, os seus códigos-fontes foram adaptados para utilização dos dados de opiniões. Na sequência realizou-se uma série de experimentações com diferentes configurações e parametrizações, as quais evidenciaram que a ferramenta não atinge bons resultados para os conjuntos de opiniões, com acurácia e precisão próximos de 60%. Além disso, através de contato com os autores do trabalho correlato que implementaram a ADN foi verificada a utilização do classificador implementado por Ruslan Salakhutdinov e Geoffrey Hinton [6] para as experimentações com DBNs, sendo esta uma ferramenta estável e testada.

Optou-se então pela adaptação deste mesmo classificador, que implementa os conceitos apresentados no Capítulo 2, fazendo uso de RBMs empilhadas, passando pelas etapas de *Pre-training* e *Fine-tuning*, em uma arquitetura de *Deep Learning*. Esta implementação também foi concebida para o Matlab utilizando como exemplo a base de dados MNIST.

Juntamente com a facilidade da ferramenta ser desenvolvida para o Matlab, onde tem-se os dados já pré-processados e transformados, ressalta-se que esta mesma implementação foi utilizada como base no trabalho de Zhou, Chen e Wang (2010), do qual os resultados obtidos são destacados como *benchmark* para a investigação realizada no presente trabalho.

4.2.3 Implementação

Após o download e instalação do pacote com o referido classificador, este foi adaptado para os conjuntos de treinamento e teste lidos pela ferramenta para os dados de opiniões, além de revisar pontos específicos da implementação. Destes, destaca-se as alterações nas camadas de entrada e saída do código original, sendo que no domínio de opiniões, tem-se 2 neurônios na camada de saída (quando abordada a questão de polaridade), em vez dos 10 neurônios quando

[3] http://www.deeplearning.net/software/theano
[4] https://github.com/rasmusbergpalm/DeepLearnToolbox
[5] https://www.python.org/
[6] http://www.cs.toronto.edu/ hinton/MatlabForSciencePaper.html

utilizada a base MNIST.

A implementação foi modificada também para receber parâmetros: o primeiro indica a quantidade de termos que será abordada para a experimentação, o segundo e o terceiro correspondem às quantidades de épocas de *Pre-training* e *Fine-tuning*, respectivamente, e o quarto é uma matriz que define o número de neurônios que cada uma das três camadas ocultas (RBMs) possui (com a função criada exibida no Apêndice E). Enquanto que na implementação original do classificador de Ruslan Salakhutdinov e Geoffrey Hinton há três camadas ocultas compostas por 500, 500 e 2000 neurônios, nas experimentações são analisadas as variações dos resultados considerando mudanças nos parâmetros mencionados.

4.2.4 Resultados

No código-fonte original da implementação adotada são utilizadas 50 épocas na etapa de treinamento, tanto na *Pre-training* quanto na *Fine-tuning*, da mesma forma que em Salakhutdinov e Hinton (2009a), sendo que com estas quantidades de épocas se superou 70% de acurácia (para o conjunto de filmes) em poucos experimentos.

Partindo-se da configuração de Zhou, Chen e Wang (2010), com as camadas ocultas possuindo 100, 100 e 200 neurônios, incrementou-se proporcionalmente e gradativamente esta configuração, buscando identificar aquela que obtém a maior acurácia. Em experimentos preliminares verificou-se que o acréscimo no número de épocas, partindo-se de 30 (também aplicado no trabalho relacionado ADN), contribui para a obtenção de um melhor resultado. Esta constatação se deu na realização do treinamento com 90 épocas, comparando-se com o aumento para 110 épocas.

A partir disso foram realizados experimentos com 120 épocas (acréscimo de 1/3 à quantidade de 90), tanto na etapa de *Pre-training* quanto na de *Fine-tuning*, e posteriormente aumentou-se proporcionalmente para 160. Os resultados para as bases de dados de filmes, GPS, câmeras e livros, após 120 épocas de treinamento são exibidos nas Tabelas 5, 6, 7 e 8, respectivamente, não sendo considerados os conjuntos com 30, 50, 100 e 200 termos, devido ao menor volume de informações, e 20000 termos, pelo alto custo computacional, buscando também evitar um elevado número de experimentações .

Tabela 5: Acurácia (%) - filmes - 120 épocas *Pre-training* e *Fine-tuning*

Camadas	Número de Termos						
1ª, 2ª, 3ª	300	500	1000	2000	3000	4000	5000
100, 100, 200	78,7	75,2	74,7	74,5	74,6	74,3	74,5
200, 200, 400	81,3	81,4	81,5	80,4	78,5	77,1	76,5
300, 300, 600	80,8	81,1	81,8	81,7	81,7	81,7	81,4
400, 400, 800	77,2	79,8	81,2	81,8	81,3	81,5	81,3
500, 500, 1000	77,2	79,8	81,2	81,7	81,7	81,4	81,2

Tabela 6: Acurácia (%) - GPS - 120 épocas *Pre-training* e *Fine-tuning*

Camadas	Número de Termos						
1ª, 2ª, 3ª	300	500	1000	2000	3000	4000	5000
100, 100, 200	69,6	69,0	69,8	71,8	73,2	73,5	73,2
200, 200, 400	68,7	69,4	68,5	68,7	68,8	68,7	68,6
300, 300, 600	69,5	69,6	69,2	69,6	69,2	69,0	69,1
400, 400, 800	68,9	68,8	68,3	69,3	69,1	68,7	69,0
500, 500, 1000	68,0	68,5	68,6	69,0	68,9	68,6	69,3

Tabela 7: Acurácia (%) - câmeras - 120 épocas *Pre-training* e *Fine-tuning*

Camadas	Número de Termos						
1ª, 2ª, 3ª	300	500	1000	2000	3000	4000	5000
100, 100, 200	84,6	84,6	84,8	84,3	84,5	84,6	84,5
200, 200, 400	84,6	84,7	84,4	84,5	84,5	84,8	84,8
300, 300, 600	84,2	84,1	84,6	84,1	84,0	84,1	84,0
400, 400, 800	84,2	84,0	83,7	83,9	83,6	83,7	84,0
500, 500, 1000	83,4	83,6	83,6	83,7	84,0	83,6	83,6

Tabela 8: Acurácia (%) - livros - 120 épocas *Pre-training* e *Fine-tuning*

Camadas	Número de Termos						
1ª, 2ª, 3ª	300	500	1000	2000	3000	4000	5000
100, 100, 200	76,9	77,5	77,6	76,6	75,4	75,6	74,0
200, 200, 400	77,1	77,5	77,3	76,4	76,3	75,5	75,4
300, 300, 600	76,4	75,9	76,4	76,5	76,3	76,0	76,0
400, 400, 800	76,0	75,9	76,1	75,7	75,8	75,7	75,2
500, 500, 1000	75,4	75,6	76,0	76,3	75,6	75,1	75,1

Analisando os resultados é constatado que para todos os conjuntos, exceto GPS, quando o número de termos é maior, os melhores resultados são obtidos em camadas com quantidades de neurônios maiores que a configuração mínima. Em contrapartida, um número menor de termos obtém maior acurácia em configurações com menos neurônios nas camadas ocultas.

Partindo para os experimentos com 160 épocas, na Tabela 9 são exibidos os valores de acurácia obtidos para o conjunto de filmes, com o modelo demonstrando uma evolução nesta configuração, com todas as diferentes quantidades de termos superando os resultados atingidos com 120 épocas de treinamento (embora a diferença da acurácia entre as configurações não seja estatisticamente significativa).

A configuração com 160 épocas para *Pre-training* e *Fine-tuning* foi aplicada também aos demais conjuntos. A Tabela 10 exibe a acurácia alcançada nos experimentos para as opiniões de câmeras, demonstrando que apenas quando adotados 3000 termos o resultado até então obtido com 120 épocas foi superado, atingindo o valor mais alto de acurácia para o conjunto.

Já a Tabela 11 mostra os resultados dos experimentos com 160 épocas de treinamento para as opiniões de livros, na qual destaca-se a maior acurácia obtida para o conjunto, 77,7% com

Tabela 9: Acurácia (%) - filmes - 160 épocas *Pre-training* e *Fine-tuning*

Camadas	Número de Termos						
1ª, 2ª, 3ª	300	500	1000	2000	3000	4000	5000
100, 100, 200	79,0	75,9	75,2	75,5	75,2	75,2	75,3
200, 200, 400	81,6	81,9	82,0	80,8	80,4	77,3	76,3
300, 300, 600	81,1	82,2	82,4	82,8	82,4	82,3	82,6
400, 400, 800	80,7	81,8	82,1	82,7	82,4	82,1	82,5
500, 500, 1000	79,6	81,1	82,1	82,3	82,4	82,7	82,2

Tabela 10: Acurácia (%) - câmeras - 160 épocas *Pre-training* e *Fine-tuning*

Camadas	Número de Termos						
1ª, 2ª, 3ª	300	500	1000	2000	3000	4000	5000
100, 100, 200	84,2	84,0	84,6	84,3	84,9	84,3	84,4
200, 200, 400	84,2	84,4	84,4	84,4	84,4	84,5	84,8
300, 300, 600	84,4	84,2	84,2	84,5	84,2	84,6	84,2
400, 400, 800	84,5	84,4	84,4	84,1	84,3	84,1	84,3
500, 500, 1000	83,8	84,0	84,4	84,2	83,4	84,2	83,9

500 termos na configuração de 200, 200 e 400 neurônios nas camadas ocultas. Todavia, a me- lhora ocorreu somente para alguns casos, e esta não foi expressiva, visto que esta configuração demanda um maior custo de processamento, posteriormente analisado.

Tabela 11: Acurácia (%) - livros - 160 épocas *Pre-training* e *Fine-tuning*

Camadas	Número de Termos						
1ª, 2ª, 3ª	300	500	1000	2000	3000	4000	5000
100, 100, 200	77,0	77,4	76,8	75,8	75,9	74,9	75,3
200, 200, 400	77,3	77,7	76,7	76,3	76,3	75,6	75,6
300, 300, 600	77,1	76,7	76,5	76,5	75,8	76,4	75,8
400, 400, 800	76,0	76,7	76,4	76,6	76,6	75,9	76,0
500, 500, 1000	76,2	76,6	76,2	76,5	76,2	76,5	75,4

Na Tabela 12 são exibidos os resultados atingidos para o conjunto de GPS, com a acurácia máxima até então obtida sendo apenas superada nos conjuntos de 1000, 2000 e 5000 termos, dos quais apenas para o primeiro o aumento foi representativo, superando em 3%.

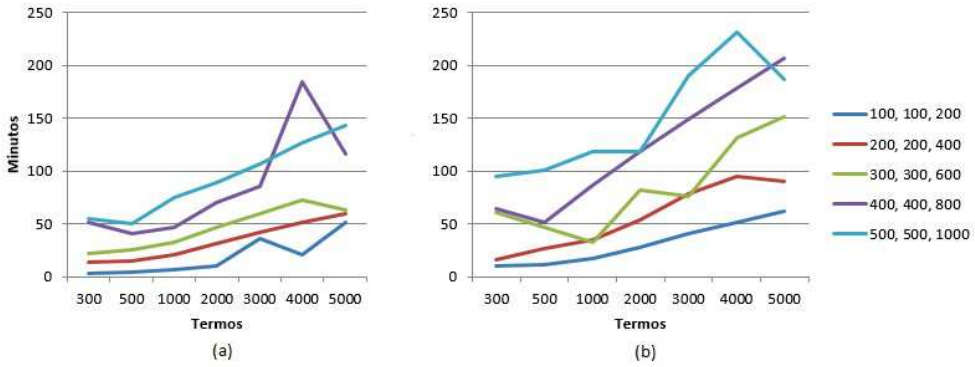
Tabela 12: Acurácia (%) - GPS - 160 épocas *Pre-training* e *Fine-tuning*

Camadas	Número de Termos						
1ª, 2ª, 3ª	300	500	1000	2000	3000	4000	5000
100, 100, 200	69,5	69,3	73,1	73,0	73,0	72,2	73,3
200, 200, 400	69,0	69,3	69,3	68,3	68,4	68,3	68,4
300, 300, 600	69,2	68,9	69,2	68,8	68,8	69,1	69,1
400, 400, 800	68,9	68,8	68,3	69,3	69,1	68,7	69,0
500, 500, 1000	68,2	67,6	67,7	67,9	67,8	68,2	67,6

Desta forma, é constatado que para o conjunto de GPS o acréscimo de épocas também não produziu resultados satisfatórios, pois além da acurácia não ter superado os valores até então obtidos (na maioria dos casos), a experimentação com 160 épocas demanda um tempo muito maior de processamento. Neste ponto destaca-se que para experimentações com ambas configurações de épocas, a com menor quantidade de neurônios nas camadas ocultas (100, 100 e 200), e consequente menor tempo de processamento, obteve os melhores resultados para as diferentes quantidades de termos.

Analisando os resultados considerando todos os conjuntos de dados, conclui-se que os resultados obtidos com 120 épocas são competitivos aos atingidos com 160, visto o menor custo computacional e tempo de processamento. Para ilustrar o quão significativa é a diferença de tempo de processamento para as duas configurações, a Figura 13 apresenta os tempos de execução dos experimentos para 120 e 160 épocas de treinamento para o conjunto de filmes. Destaca-se o gradativo aumento do custo conforme o acréscimo do volume de dados, visto a variação dos valores extremos (300 termos, com a menor quantidade de neurônios e 5000 termos para o máximo de neurônios, apesar de esta ter demandado um tempo ligeiramente menor que com 4000 termos quando utilizadas 160 épocas).

Figura 13: Tempo de execução - 120 (a) e 160 (b) épocas *Pre-training* e *Fine-tuning*



A acurácia foi a unidade de medida com maior enfoque, visto a sua utilização nos trabalhos relacionados. Entretanto, outras unidades são registradas para os experimentos, como o *recall* e precisão, sendo estas exibidas nas Figuras 14 e 15, respectivamente, para o conjunto de filmes.

Verifica-se que para a classe *positiva* o valor de *recall* decai conforme aumenta o número de termos para as configurações com menor quantidade de neurônios, enquanto que a precisão seguiu o mesmo padrão apenas para a configuração com 300, 300 e 600 neurônios e para a classe *positiva* com 200, 200 e 400 neurônios.

As Figuras 16 e 17 apresentam o *recall* e a precisão atingidos nas experimentações realizadas com as três configurações com melhores resultados: 100, 100 e 200; 200, 200 e 400 e 300, 300 e 600 neurônios nas camadas ocultas, para o conjunto de GPS. Assim como para as opiniões sobre filmes, o modelo seguiu um comportamento semelhante alternando a quantidade de épocas: para a classe *positiva*, com 100, 100 e 200 neurônios nas camadas ocultas atingiu precisão próxima das demais configurações, até 500 termos, elevando-se bastante a partir de

Figura 14: *Recall* (%) p/ filmes com 120 (a) e 160 (b) épocas *Pre-training* e *Fine-tuning*

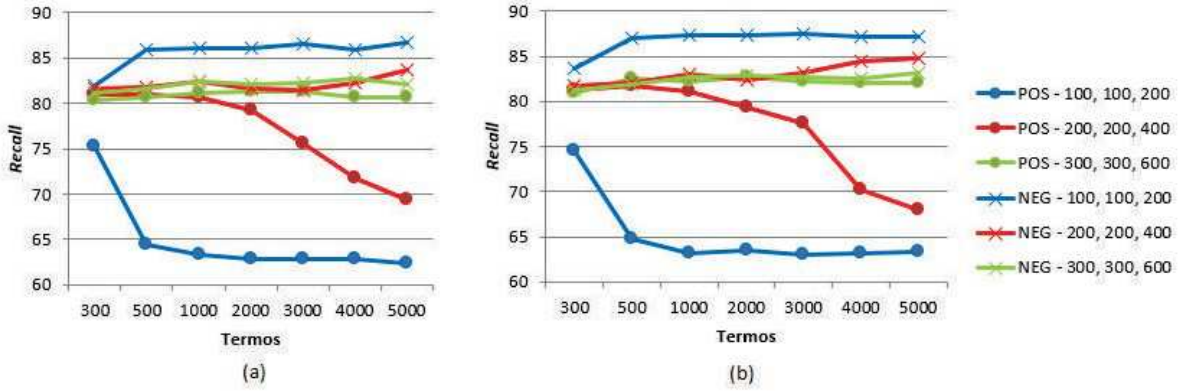


Figura 15: Precisão (%) p/ filmes com 120 (a) e 160 (b) épocas *Pre-training* e *Fine-tuning*

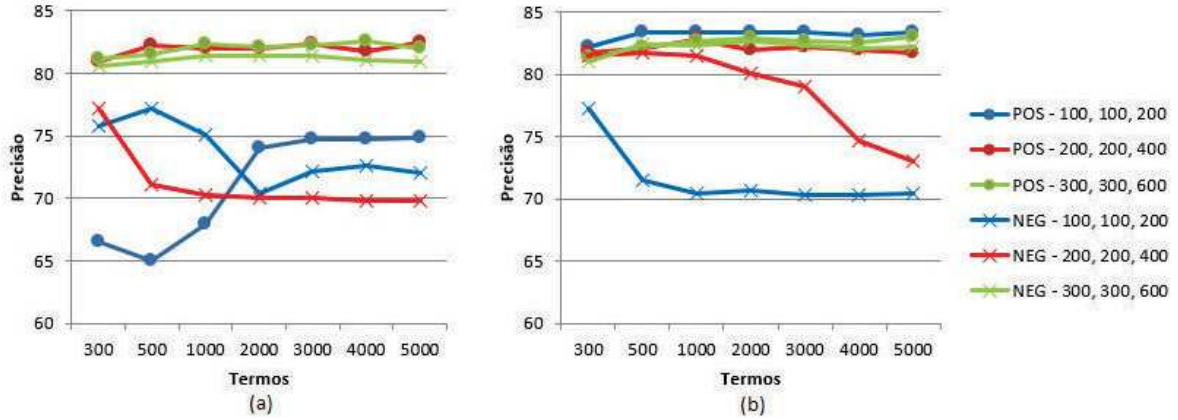
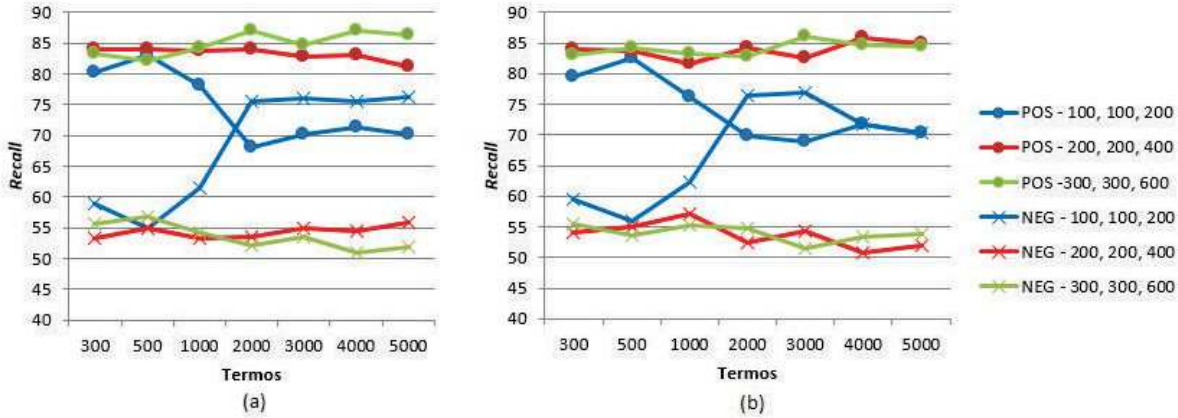


Figura 16: *Recall* (%) p/ GPS com 120 (a) e 160 (b) épocas *Pre-training* e *Fine-tuning*



1000 termos. No entanto, diferentemente do conjunto de filmes, o *recall* foi maior para a classe

positiva na maioria dos casos.

Já as Figuras 18 e 19 exibem o *recall* e a precisão, respectivamente, atingidos para o conjunto

Figura 17: Precisão (%) p/ GPS com 120 (a) e 160 (b) épocas *Pre-training* e *Fine-tuning*

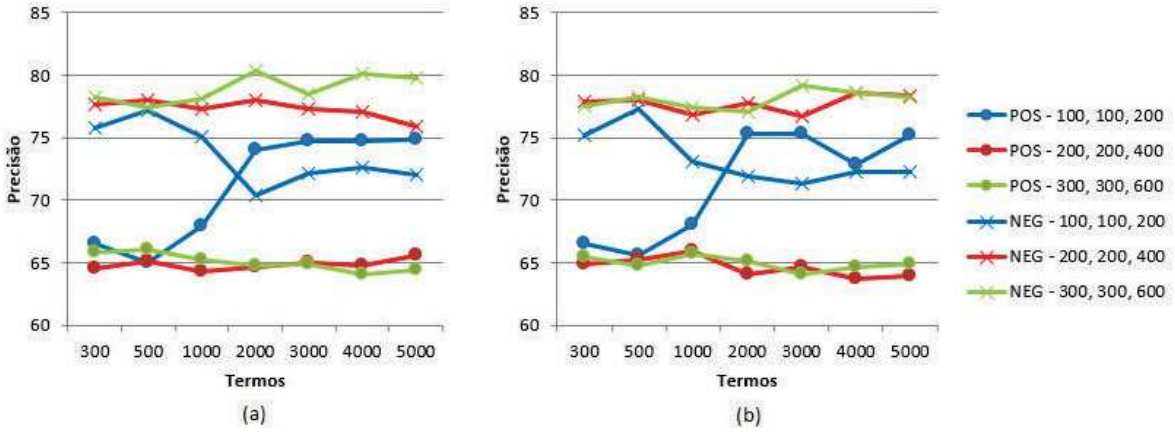


Figura 18: *Recall* (%) p/ câmeras com 120 (a) e 160 (b) épocas *Pre-training* e *Fine-tuning*

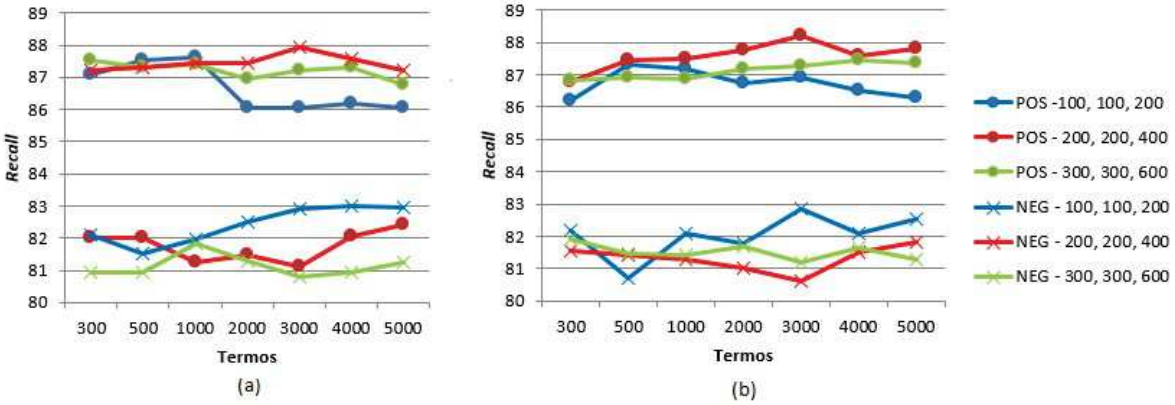
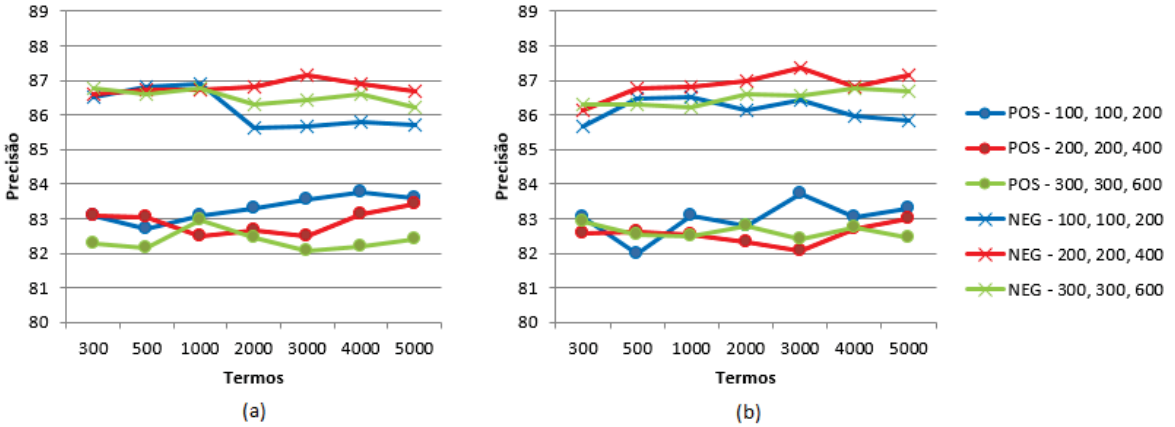


Figura 19: Precisão (%) p/ câmeras com 120 (a) e 160 (b) épocas *Pre-training* e *Fine-tuning*



de câmeras, no qual o comportamento foi distinto na mudança de épocas, quando utilizada a configuração 200, 200 e 400 neurônios nas camadas ocultas, com a classe *positiva* obtendo *recall* superior à *negativa* na configuração com 120 épocas e inferior quando realizadas 160

épocas. A precisão demonstrou uma tendência semelhante para todas as configurações, com a classe *negativa* obtendo valores superiores à *positiva*.

Para as opiniões referentes a livros, o modelo obteve comportamento semelhante considerando um diferente número de épocas, com o *recall* exibido na Figura 20, em que são verificados valores maiores para a classe *positiva*. Em contrapartida, a precisão também seguiu comportamento semelhante alternando o número de épocas, mas com a classe *negativa* obtendo valores superiores, conforme apresentado na Figura 21.

Figura 20: *Recall* (%) p/ livros com 120 (a) e 160 (b) épocas *Pre-training* e *Fine-tuning*

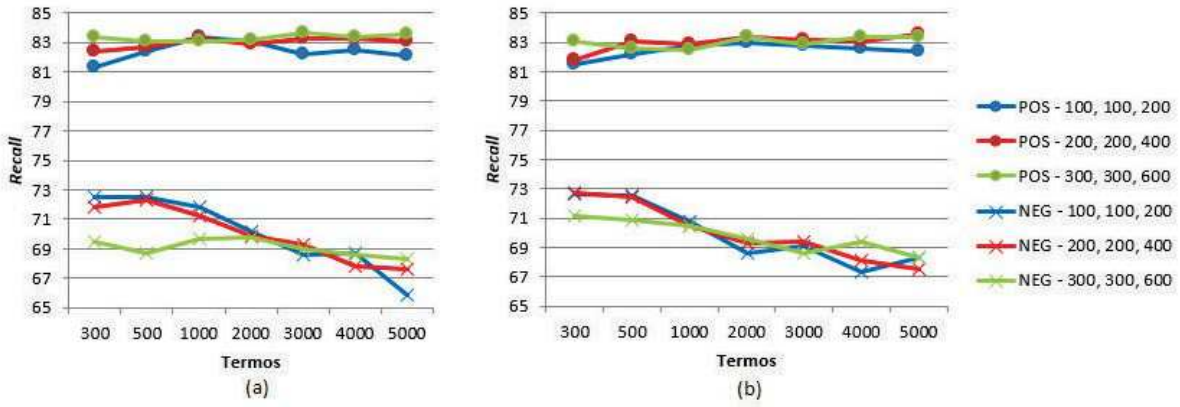
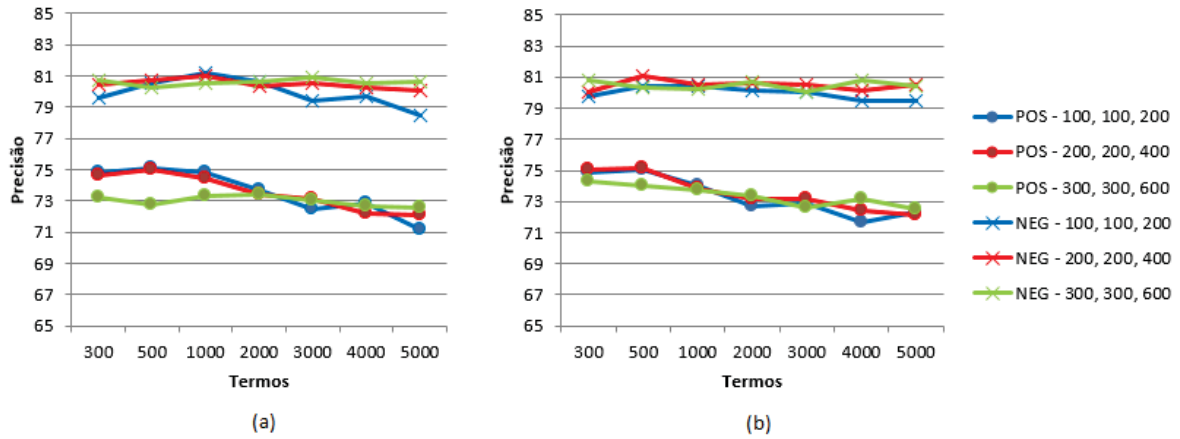


Figura 21: Precisão (%) p/ livros com 120 (a) e 160 (b) épocas *Pre-training* e *Fine-tuning*



A unidade de medida *f-measure*, calculada em função de *recall* e precisão, também foi computada e é exibida na Figura 22 aquela obtida para filmes, onde destaca-se a considerável queda identificada conforme aumenta a quantidade de termos nas configurações com menor número de neurônios nas camadas ocultas.

A Figura 23 apresenta a *f-measure* calculada para os experimentos com as opiniões de GPS, onde comprova-se a grande diferença de performance do modelo para os conjuntos que superam

Figura 22: *F-measure* (%) p/ filmes com 120 (a) e 160 (b) épocas *Pre-training* e *Fine-tuning*

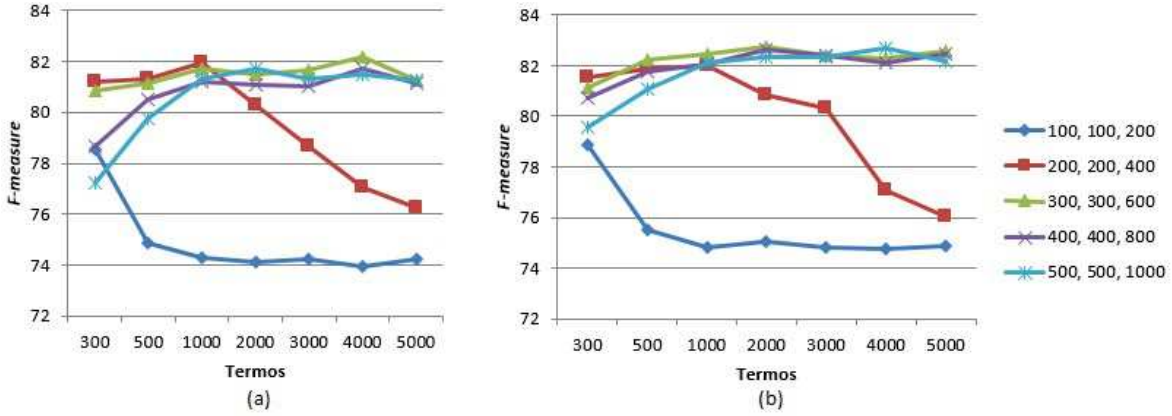
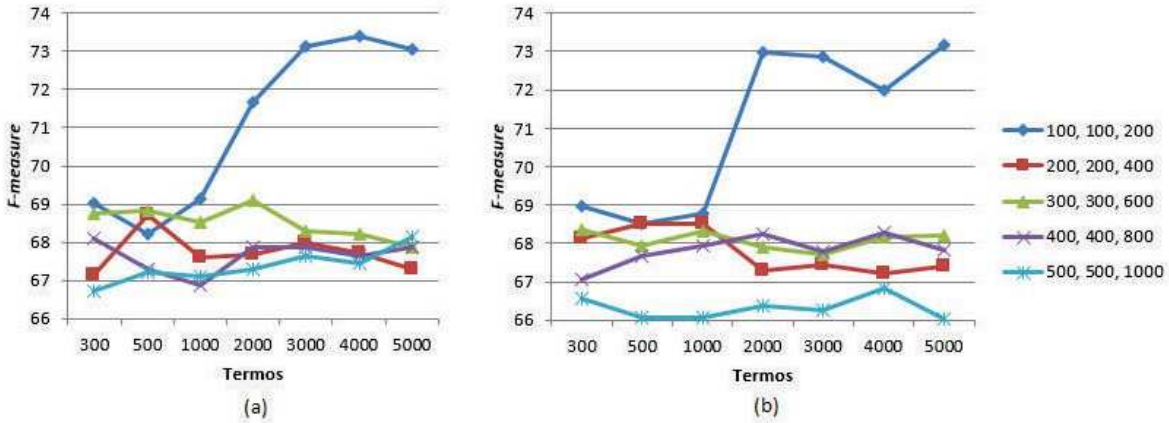


Figura 23: *F-measure* (%) p/ GPS com 120 (a) e 160 (b) épocas *Pre-training* e *Fine-tuning*



1000 termos, quando aplicada a configuração com menor quantidade de neurônios nas camadas ocultas.

Figura 24: *F-measure* (%) p/ câmeras com 120 (a) e 160 (b) épocas *Pre-training* e *Fine-tuning*

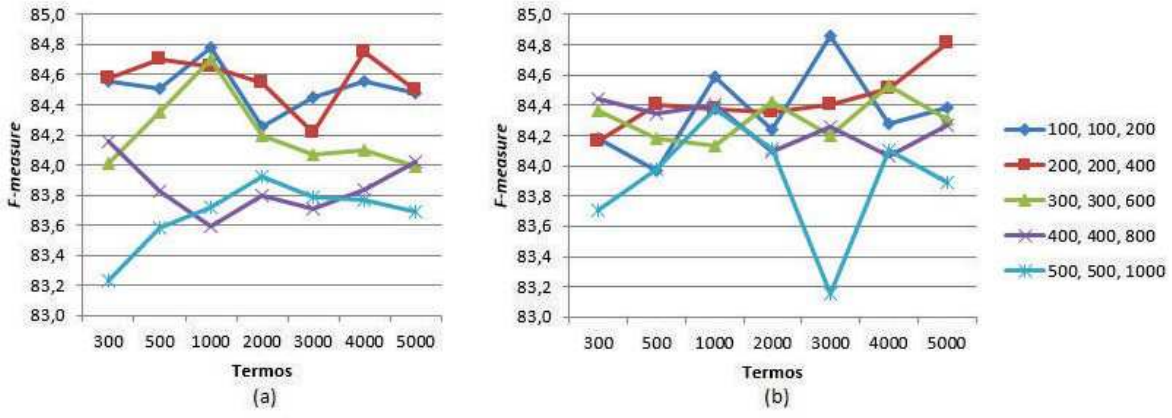
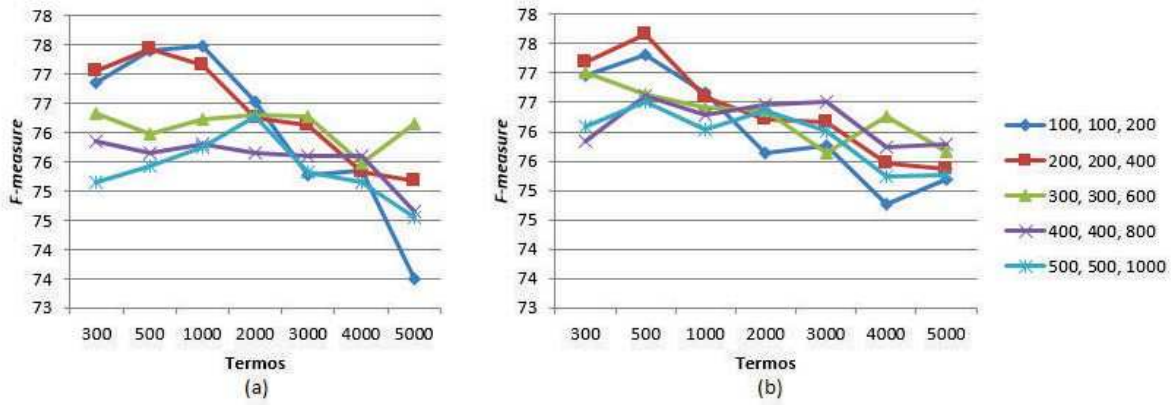


Figura 25: *F-measure* (%) p/ livros com 120 (a) e 160 (b) épocas *Pre-training* e *Fine-tuning*



Por fim, a Figura 24 exibe a *f-measure* para as opiniões de câmeras, onde os valores mais altos foram obtidos quando adotados 100, 100 e 200 e 200, 200 e 400 neurônios nas camadas ocultas e os mais baixos na configuração com 500, 500 e 1000 neurônios; e a Figura 25 exibe a *f-measure* para o conjunto de livros, onde as experimentações atingiram valores próximos para as diferentes configurações de épocas.

Analisando os resultados obtidos até então, conclui-se que o acréscimo do número épocas não surtiu efeito significativo. Por isso, novamente foram realizados experimentos com 120 épocas de treinamento na etapa de *Fine-tuning*, mas alterando a quantidade de épocas na etapa *Pre-training*, adotando-se a configuração de 30 épocas, mesma parametrização abordada em Zhou, Chen e Wang (2010).

Foram também efetuados experimentos utilizando 50 épocas na etapa de *Pre-training*, seguindo Salakhutdinov e Hinton (2009a), e também com 10 épocas, visando analisar o comportamento do modelo e tentar identificar a melhor configuração para a etapa de *Pre-training*. Após esta análise constatou-se que 10 é um número insuficiente de épocas, e que 50 não gerou resultados significativamente melhores, mantendo-se então as 30 épocas para todos os conjuntos.

Esta ação de diminuição do total de épocas na etapa de *Pre-training* foi adotada para diminuir a demanda por processamento e para evitar que um excessivo número de épocas em um treinamento não-supervisionado pudesse perder informações do conteúdo dos dados. A Tabela 13 apresenta os resultados obtidos com a referida configuração para o conjunto de filmes, em que verifica-se que a acurácia até então atingida não foi superada em nenhum caso.

Já para as opiniões referentes a GPS, constata-se que a alteração na parametrização surtiu efeito, visto que os resultados previamente obtidos foram superados na maioria dos casos, conforme os valores apresentados na Tabela 14, destacando ainda a obtenção dos melhores resultados para o conjunto, superando pela primeira vez os 74% de acurácia para esta base (4000 termos na configuração com 100, 100 e 200 neurônios nas camadas ocultas).

Para o conjunto de câmeras, a acurácia obtida na configuração com número de épocas de

Tabela 13: Acurácia (%) - filmes - 30 épocas *Pre-training* e 120 épocas *Fine-tuning*

Camadas	Número de Termos						
1ª, 2ª, 3ª	300	500	1000	2000	3000	4000	5000
100, 100, 200	79,1	75,1	75,0	74,3	74,2	73,8	73,9
200, 200, 400	81,1	81,6	81,5	80,6	79,1	77,2	76,5
300, 300, 600	80,6	81,0	81,3	81,3	81,3	81,5	81,4
400, 400, 800	80,0	80,6	81,4	81,4	81,5	81,5	81,7
500, 500, 1000	78,3	80,0	80,8	81,8	81,1	81,0	81,5

Tabela 14: Acurácia (%) - GPS - 30 épocas *Pre-training* e 120 épocas *Fine-tuning*

Camadas	Número de Termos						
1ª, 2ª, 3ª	300	500	1000	2000	3000	4000	5000
100, 100, 200	70,3	68,8	69,7	74,8	73,5	74,1	73,9
200, 200, 400	70,1	69,9	69,4	68,7	69,1	68,3	69,2
300, 300, 600	69,2	69,8	70,1	69,3	69,3	69,1	69,3
400, 400, 800	68,9	69,5	69,6	69,5	69,8	69,3	69,4
500, 500, 1000	68,5	68,4	69,7	69,0	69,6	69,1	69,1

e 120, para *Pre-training* e *Fine-tuning*, respectivamente, é exibida na Tabela 15. Os resultados com a mesma configuração para as opiniões de livros são demonstrados na Tabela 16.

Tabela 15: Acurácia (%) - câmeras - 30 épocas *Pre-training* e 120 épocas *Fine-tuning*

Camadas	Número de Termos						
1ª, 2ª, 3ª	300	500	1000	2000	3000	4000	5000
100, 100, 200	85,1	84,7	84,8	84,6	84,4	84,6	84,4
200, 200, 400	84,7	85,1	84,6	84,8	84,5	85,2	84,4
300, 300, 600	84,6	84,6	84,3	84,1	84,4	84,4	84,1
400, 400, 800	84,2	83,8	84,0	84,3	84,2	84,1	84,0
500, 500, 1000	83,7	84,0	84,0	84,0	84,1	84,0	83,7

Tabela 16: Acurácia (%) - livros - 30 épocas *Pre-training* e 120 épocas *Fine-tuning*

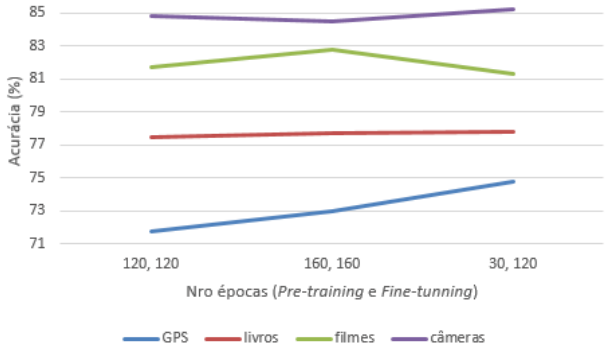
Camadas	Número de Termos						
1ª, 2ª, 3ª	300	500	1000	2000	3000	4000	5000
100, 100, 200	77,3	77,6	77,2	76,3	75,8	75,1	76,0
200, 200, 400	77,0	77,8	77,2	76,4	76,6	75,8	75,5
300, 300, 600	76,6	76,7	76,7	77,2	76,4	76,5	75,9
400, 400, 800	75,9	76,4	76,3	76,1	75,6	76,2	75,7
500, 500, 1000	76,0	75,9	76,1	75,5	75,7	75,7	74,9

Assim como para GPS, os resultados foram satisfatórios, visto o grande número de casos de acurácia maior e a obtenção dos valores extremos para o conjunto, onde para câmeras foi superado os 85% para 500 termos (nas configurações de 100, 100 e 200; e 200, 200 e 400 neurônios)

e para livros foi atingida a acurácia máxima dos experimentos, 77,8%, quando utilizados 500 termos com 200, 200 e 400 neurônios nas camadas ocultas.

A Figura 26 apresenta as melhores configurações obtidas para GPS (2000 termos e camadas ocultas com 100, 100 e 200 neurônios), livros (500 termos e camadas ocultas com 200, 200 e 400 neurônios), câmeras (4000 termos e camadas ocultas com 200, 200 e 400 neurônios) e filmes (2000 termos e camadas ocultas com 300, 300 e 600 neurônios), considerando a variação no número de épocas empregado.

Figura 26: Variação das melhores configurações



Com os resultados obtidos com o diferente número de épocas para o treinamento nas etapas de *Pre-training* e *Fine-tuning*, constata-se que experimentações devem ser realizadas considerando esta alternância, visto que a acurácia até então atingida foi superada para a maioria dos casos, exceto para o conjunto de filmes, onde não ocorreu melhora em nenhum caso.

Apesar de um número menor de épocas na etapa de *Pre-training* possibilitar que o modelo não perca características encontradas nos dados através de um número excessivo de épocas, o conjunto pode possuir dados tão esparsos a ponto de um número menor de épocas não ser o suficiente para a identificação destas características.

Destaca-se que a configuração com menor número de épocas, além de obter uma acurácia melhor para a maioria dos conjuntos, possui um custo menor de processamento, conforme a Figura 27 (comparando-a com a Figura 13), que demonstra o tempo de execução na configuração com diferentes quantidades de épocas nas etapas de *Pre-training* e *Fine-tuning* para o conjunto de filmes.

A diferença de performance acaba justificando a utilização desta última configuração, visto que os resultados foram estatisticamente significativos, o que foi comprovado através de testes de significância, considerando um nível de 5% para os tempos de execução com os diferentes números de épocas. Os mesmos testes de significância foram realizados analisando a acurácia obtida para as diferentes configurações. No entanto, para estes a variação da acurácia não foi comprovada como estatisticamente significativa.

As Tabelas 17 e 18 apresentam os valores de desvio-padrão considerados nos testes de significância (valores médios ja citados na Tabela 13 e Figura 27).

Figura 27: Tempo de execução - 30 épocas *Pre-training* e 120 *Fine-tuning*

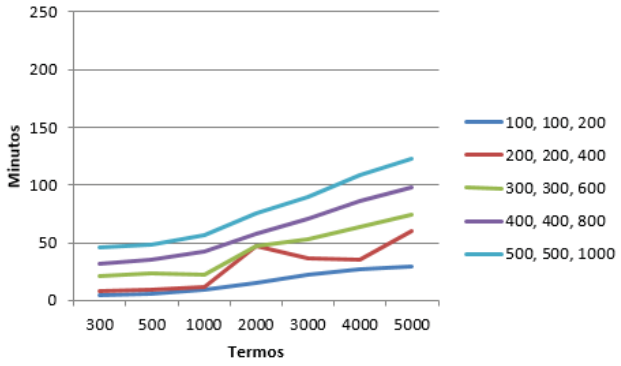


Tabela 17: Desv.padr. - Acurácia (%) - filmes - 30 épocas *Pre-training* e 120 épocas *Fine-tuning*

1ª, 2ª, 3ª	300	500	1000	2000	3000	4000	5000
400, 400, 800	0,25	0,23	0,48	0,31	0,26	0,32	0,66
500, 500, 1000	0,65	0,29	0,26	0,21	0,50	0,22	0,37

Tabela 18: Desv.padr. - Minutos de Exec. - filmes - 30 épocas *Pre-training* e 120 épocas *Fine-tuning*

Camadas	Número de Termos						
1ª, 2ª, 3ª	300	500	1000	2000	3000	4000	5000
400, 400, 800	1,00	1,15	0,58	2,08	1,53	2,65	2,08
500, 500, 1000	1,53	1,00	2,08	2,31	2,65	3,46	4,62

4.3 Considerações

Neste capítulo foi apresentada a modelagem abordada, com a possibilidade das etapas de pré-processamento e Mineração de Dados descritas serem replicadas em diferentes cenários. Foram também relatadas as experimentações realizadas e os resultados obtidos.

Analisando os trabalhos relacionados, se verifica em Zhou, Chen e Wang (2010) a utilização de 10000 termos selecionados considerando sua frequência, em uma configuração com um menor número de neurônios nas camadas ocultas (comparado com o presente trabalho), sem relatar impactos de variações nesta configuração. Com o modelo proposto de ADN, que combina um método de aprendizado supervisionado (*Active Learning*), juntamente com uma estrutura *Deep Learning*, obteve-se 76,3% de acurácia na base de filmes de Pang e Lee (2004), enquanto que na utilização da DBN de Ruslan Salakhutdinov e Geoffrey Hinton (sem o algoritmo de *Active Learning*) foi atingida a acurácia de 71,5%.

Em extensão ao modelo de ADN (ZHOU; CHEN; WANG, 2013), foi atingida uma acurácia maior, 76,4%, mas em função de melhorias no procedimento de *Active Learning* proposto (IADN). No trabalho de Glorot, Bordes e Bengio (2011a), com o modelo de *Rectifier Neural Network* (RNN), utilizando diferentes funções de ativação, mas com etapas de pré-processamento semelhantes ao modelo ADN, obteve-se o valor de 78,95%. Este corresponde

à média da acurácia obtida para os quatro conjuntos adotados no trabalho relacionado, o que pode parecer uma forma não muito adequada para serem avaliados resultados uma vez que os domínios de opiniões são diversos. Desta forma, como os conjuntos de dados utilizados na presente investigação são diferentes do trabalho correlato, é impossibilitada uma comparação precisa.

A Tabela 19 compara a acurácia máxima obtida nos trabalhos mencionados com a atingida no presente trabalho. Se verifica que mesmo com uma quantidade menor de termos a acurácia foi superada. Como *benchmark* ainda tem-se a acurácia obtida através da submissão dos dados de opiniões referentes a filmes, na sua forma bruta, para a AlchemyAPI, ferramenta que permite a avaliação de textos por parte de sua base de conhecimento através de uma conexão http, atingindo 77,8% de acurácia.

Tabela 19: Acurácia (%) - Comparação com trabalhos relacionados

	DBN	ADN	IADN	*Alc. API*	Pres. Trab.
Filmes	71,3	76,3	76,4	77,8	82,8
Livros	64,3	69,0	69,7	-	77,8

Na literatura são encontrados trabalhos que utilizam os mesmos conjuntos de dados atingindo acurácia superior, sem o emprego de *Deep Learning*, conforme comparação exibida na Tabela 20, onde constam os valores obtidos no presente trabalho juntamente com a acurácia máxima atingida no trabalho de Moraes (2013). Enquanto isso, modelos que aplicaram *Deep Learning*, como Glorot, Bordes e Bengio (2011a), Zhou, Chen e Wang (2010) e Zhou, Chen e Wang (2013) atingiram resultados na faixa dos 70%, sendo que nos resultados das experimentações realizadas com os dados refinados foram superados os 80% de acurácia.

Tabela 20: Acurácia (%) - Comparação com trabalho Moraes (2013)

Dados	Moraes (2013)	Pres. Trab.
GPS	87,3	74,8
Câmeras	90,3	85,2
Filmes	86,5	82,8
Livros	81,8	77,8

Embora a utilização de *Deep Learning* para Mineração de Opiniões, em um primeiro momento, não supere trabalhos já observados na literatura, abrem-se possibilidades de sua exploração nesta área, visto que *Deep Learning* tem demonstrado resultados satisfatórios em várias outras áreas. No entanto, as abordagens em Mineração de Opiniões não passaram por um refinamento mais apurado na etapa de pré-processamento, como o apresentado no presente trabalho, deixando apenas *Deep Learning* para seleção de características, em dados mais brutos. Desta forma, verifica-se que por mais robusta e eficiente que a técnica seja na sua capacidade de extração de características, outras técnicas podem contribuir para um melhor resultado, possibilitando que esta extração ocorra em dados já refinados.

5 CONCLUSÃO

Estruturas com múltiplas camadas têm obtido evolução em seus resultados desde o trabalho de Hinton, Osindero e Teh (2006) e têm estado em evidência, sendo alvo de estudos inclusive de grandes corporações. Dado os supostos promissores resultados atingidos por *Deep Learning* em diferentes áreas, como reconhecimento de imagens e áudio, tem-se realizados estudos que aplicam a técnica em dados textuais, destacando a abordagem em Mineração de Opiniões.

Estudos que buscam identificar o sentimento contido em opiniões têm seu desenvolvimento favorecido pelo contínuo crescimento do volume de informações referentes a opiniões, juntamente com o ambiente colaborativo da Web 2.0 propiciando a emissão dessas opiniões e a necessidade das empresas compreenderem seus consumidores. Desta forma surgem mais trabalhos em Mineração de Opiniões, abordada por diferentes áreas.

Quando o estudo para a identificação de polaridade de opiniões envolve a técnica de *Deep Learning*, esta é empregada para uma das suas principais atribuições: seleção de características, indo ao encontro dos princípios da técnica (BENGIO, 2009). O presente trabalho demonstrou que a combinação de dados com o emprego de um processo de refinamento de dados brutos, efetuando-se etapas no estágio de pré-processamento e a seleção de termos por IG, e *Deep Learning* (aplicado para extração de características em dados refinados), proporciona a obtenção de melhores resultados, destacando a superação em 6% da acurácia atingida nos trabalhos relacionados para o conjunto de dados de opiniões sobre filmes de Pang e Lee (2004).

Por outro lado, foi constatada a necessidade de experimentações com inúmeras configurações, com o presente trabalho buscando averiguar o impacto que determinados parâmetros podem causar no comportamento do classificador. Além da avaliação desta parametrização, o estudo se diferenciou dos trabalhos relacionados no cálculo e apresentação de diferentes métricas (não somente a acurácia), possibilitando uma análise mais detalhada dos resultados obtidos. Isto pode ser constatado através das Figuras 14 a 21, em que se identifica um distinto comportamento entre as classes *positiva* e *negativa*.

Baseado nestes resultados, verificou-se que os conjuntos com diferentes quantidades de termos demonstraram comportamentos distintos na fase de experimentação, onde se buscou analisar as melhores configurações para cada conjunto. As diferenças ocorreram principalmente em relação ao número de neurônios nas três camadas ocultas.

Já alternando o número de épocas, o comportamento mostrou-se semelhante para os conjuntos, onde o acréscimo de épocas para a etapa de treinamento obteve melhores resultados. Todavia, estes não foram tão significantes, visto o maior custo computacional gerado. Com o intuito de diminuir estes custos, mas ainda visando a obtenção de melhores resultados, foram realizados experimentos com diferentes quantidades de épocas nas etapas de *Pre-training* e *Fine-tuning*.

Os competitivos resultados atingidos, comparando-se aos trabalhos relacionados, dão perspectiva de extensão a este trabalho e continuidade às experimentações, levando em conta parâ-

metros até então não considerados (como taxa de aprendizado) ou até alteração na implementa- ção da DBN da presente modelagem.

Estas alterações podem consistir em diferentes funções de ativação, como em Glorot, Bordes e Bengio (2011a), que considerou a *softplus*; ou alterar a implementação também para analisar variações de estruturas de *Deep Learning*, como as *Deep Boltzmann Machines* (DBM) (SALAKHUTDINOV; HINTON, 2009b) e *Robust Boltzmann Machines* (TANG, 2012); ou a possibilidade de aplicar uma RBM Gaussiana na primeira camada da DBN, que conforme Salakhutdinov (2009), gera um comportamento melhor quando os valores de entrada são números reais. Uma abordagem distinta também poderia ser adotada na etapa de pré-processamento, onde os conjuntos formados a partir de diferentes quantidades de termos, poderiam ser criados a partir de valores do IG.

Uma das possibilidades de extensão ainda seria a aplicação da investigação realizada em contextos desbalanceados e tratamento do desbalanceamento por *undersampling*, visto que todos experimentos foram realizados com dados balanceados. Da mesma forma, há perspectiva de pesquisa da combinação de *Deep Learning* com outras técnicas, buscando potencializar as qualidades de cada uma, como proposto em Zhou, Chen e Wang (2014), onde foi elaborado um modelo que combina *Deep Learning* com lógica *fuzzy*, e Socher (2014), que criou a estrutura *Recursive Neural Tensor Network* (RNTN), que além da representação de sentenças por vetores de termos, os trata no formato de árvore em um modelo recursivo.

Em última análise, dado o sucesso dos algoritmos de *Deep Learning* em várias áreas, mas não superando trabalhos no estado da arte em Mineração de Opiniões, espera-se que a presente abordagem seja referência em problemas de Mineração de Opiniões com aplicação de *Deep Learning*, visto que a investigação realizada sugere que, em vez da pura utilização da técnica de *Deep Learning* na identificação de características dos dados, é possível a combinação da técnica com outros métodos que almejam o aumento da qualidade dos dados, efetuando assim a extração de características em dados refinados.

REFERÊNCIAS

ARNOLD, L. Learning Deep Representations Toward a better understanding of the deep learning paradigm. 2013. Tese de Doutorado — Laboratoire des Sciences et Mécaniques de l'Ingénieur - Université Paris Sud, Paris, 2013.

ARNOLD, L.; REBECCHI, S.; CHEVALLIER, S.; PAUGAM-MOISY, H. An introduction to deep-learning. In: ADVANCES IN COMPUTATIONAL INTELLIGENCE AND MACHINE LEARNING, ESANN'2011, 2011. Anais... [S.l.: s.n.], 2011. p. 477–488.

ATTWELL, D.; LAUGHLIN, S. B. An Energy Budget for Signaling in the Grey Matter of the Brain. Journal of Cerebral Blood Flow & Metabolism, [S.l.], v. 21, n. 10, p. 1133–1145, 2001.

BENGIO, Y. Learning Deep Architectures for AI. Hanover, MA, USA: Now Publishers Inc., 2009.

BENGIO, Y.; LAMBLIN, P.; POPOVICI, D.; LAROCHELLE, H. Greedy layer-wise training of deep networks. In: SCHöLKOPF, B.; PLATT, J.; HOFFMAN, T. (Ed.). Advances in Neural Information Processing Systems 19. Cambridge, MA: MIT Press, 2007. p. 153–160.

BERRY, M. W.; KOGAN, J. (Ed.). Text Mining: applications and theory. Chichester, UK: Wiley, 2010.

BLITZER, J.; DREDZE, M.; PEREIRA, F. Biographies, bollywood, boomboxes and blenders: domain adaptation for sentiment classification. In: IN ACL, 2007. Anais... [S.l.: s.n.], 2007. p. 187–205.

CARREIRA-PERPINAN, M. A.; HINTON, G. E. On Contrastive Divergence Learning. In: 2005. Anais... Artificial Intelligence and Statistics 2005: Barbados, 2005.

CARVALHO, A. C. P. de; BRAGA, A. P.; LUDERMIR, T. B. Redes Neurais Artificiais: teoria e aplicações. 2. ed. [S.l.]: Livros Técnicos e Científicos Editora, 2014.

DASGUPTA, S.; NG, V. Mine the Easy, Classify the Hard: a semi-supervised approach to automatic sentiment classification. In: JOINT CONFERENCE OF THE 47TH ANNUAL MEETING OF THE ACL AND THE 4TH INTERNATIONAL JOINT CONFERENCE ON NATURAL LANGUAGE PROCESSING OF THE AFNLP: VOLUME 2 - VOLUME 2, 2009, Stroudsburg, PA, USA. Proceedings... Association for Computational Linguistics, 2009. p. 701–709. (ACL '09).

DENG, L.; YU, D. DEEP LEARNING: methods and applications. [S.l.]: NOW Publishers, 2014. (MSR-TR-2014-21).

FAN, H.; CAO, Z.; JIANG, Y.; YIN, Q.; DOUDOU, C. Learning Deep Face Representation. CoRR, [S.l.], v. abs/1403.2802, 2014.

FAYYAD, U. M.; PIATETSKY-SHAPIRO, G.; SMYTH, P.; UTHURUSAMY, R. (Ed.). Advances in Knowledge Discovery and Data Mining. Menlo Park, CA, USA: American Association for Artificial Intelligence, 1996.

FELDMAN, R.; SANGER, J. Text Mining Handbook: advanced approaches in analyzing unstructured data. New York, NY, USA: Cambridge University Press, 2006.

FORMAN, G. An Extensive Empirical Study of Feature Selection Metrics for Text Classification. J. Mach. Learn. Res., [S.l.], v. 3, p. 1289–1305, Mar. 2003.

FREEMAN, J. A.; SKAPURA, D. M. Neural Networks: algorithms, applications, and programming techniques. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc., 1991.

GLOROT, X.; BORDES, A.; BENGIO, Y. Deep Sparse Rectifier Neural Networks. In: FOURTEENTH INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE AND STATISTICS (AISTATS-11), 2011a. Proceedings... [S.l.: s.n.], 2011a. p. 315–323.

GLOROT, X.; BORDES, A.; BENGIO, Y. Domain Adaptation for Large-Scale Sentiment Classification: a deep learning approach. In: ICML, 2011b. Anais... Omnipress, 2011b. p. 513–520.

GOLDSCHMIDT, R.; PASSOS, E. Data Mining: um guia prático. [S.l.]: Campus, 2005.

GONG, N.; SHAO, W.; XU, H. The Conjugate Gradient Method with neural network control. In: INTELLIGENT SYSTEMS AND KNOWLEDGE ENGINEERING (ISKE), 2010 INTERNATIONAL CONFERENCE ON, 2010. Anais... [S.l.: s.n.], 2010. p. 82–84.

HAYKIN, S. Kalman Filtering And Neural Networks. [S.l.]: Wiley Online Library,

2001. HE, Y.; LIN, C.; ALANI, H. Automatically Extracting Polarity-bearing Topics for Cross-domain Sentiment Classification. In: ANNUAL MEETING OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS: HUMAN LANGUAGE TECHNOLOGIES - VOLUME 1, 49., 2011, Stroudsburg, PA, USA. Proceedings... Association for Computational Linguistics, 2011. p. 123–131. (HLT '11).

HINTON, G. Deep Belief Nets. NIPS Tutorial. Canadian Institute for Advanced Research and Department of Computer Science University of Toronto, 2007.

HINTON, G. E. Training Products of Experts by Minimizing Contrastive Divergence. Neural Computation, [S.l.], v. 14, n. 8, p. 1771–1800, 2002.

HINTON, G. E. A Practical Guide to Training Restricted Boltzmann Machines. In: MONTAVON, G.; ORR, G. B.; MüLLER, K.-R. (Ed.). Neural Networks: tricks of the trade (2nd ed.). [S.l.]: Springer, 2012. p. 599–619. (Lecture Notes in Computer Science, v. 7700).

HINTON, G. E.; OSINDERO, S.; TEH, Y.-W. A Fast Learning Algorithm for Deep Belief Nets. Neural Comput., Cambridge, MA, USA, v. 18, n. 7, p. 1527–1554, 2006.

HINTON, G. E.; SALAKHUTDINOV, R. R. Reducing the dimensionality of data with neural networks. Science, [S.l.], v. 313, n. 5786, p. 504–507, Jul 2006.

JARRETT, K.; KAVUKCUOGLU, K.; LECUN, Y. What is the Best Multi-Stage Architecture for Object Recognition? In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS, 2009. Anais... ICCV, 2009.

JOLLIFFE, I. Principal Component Analysis. [S.l.]: Springer Verlag, 1986.

LEE, H.; PHAM, P. T.; LARGMAN, Y.; NG, A. Y. Unsupervised feature learning for audio classification using convolutional deep belief networks. In: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS 22: 23RD ANNUAL CONFERENCE ON NEURAL INFORMATION PROCESSING SYSTEMS 2009. PROCEEDINGS OF A MEETING HELD 7-10 DECEMBER 2009, VANCOUVER, BRITISH COLUMBIA, CANADA., 2009. Anais... [S.l.: s.n.], 2009. p. 1096–1104.

LI, S.; ZONG, C. Multi-domain adaptation for sentiment classification: using multiple classifier combining methods. In: PROCEDING OF THE CONFERENCE ON NATURAL LANGUAGE PROCESSING AND KNOWLEDGE ENGINEERING, 2008. Anais... [S.l.: s.n.], 2008.

LIU, B. Sentiment Analysis and Opinion Mining. [S.l.]: Morgan & Claypool Publishers, 2012. (Synthesis Lectures on Human Language Technologies).

MAAS, A. L.; DALY, R. E.; PHAM, P. T.; HUANG, D.; NG, A. Y.; POTTS, C. Learning Word Vectors for Sentiment Analysis. In: ANNUAL MEETING OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS: HUMAN LANGUAGE TECHNOLOGIES - VOLUME 1, 49., 2011, Stroudsburg, PA, USA. Proceedings... Association for Computational Linguistics, 2011. p. 142–150. (HLT '11).

MITCHELL, T. M. Machine Learning. 1. ed. New York, NY, USA: McGraw-Hill, Inc., 1997.

MNIH, V.; LAROCHELLE, H.; HINTON, G. E. Conditional Restricted Boltzmann Machines for Structured Output Prediction. CoRR, [S.l.], v. abs/1202.3748, 2012.

MORAES, R. de. Uma Investigação Empírica e Comparativa da Aplicação de RNAs ao Problema de Mineração de Opiniões e Análise de Sentimentos. 2013. Dissertação de Mestrado — Programa de Pós-Graduação em Computação Aplicada - Universidade do Vale do Rio dos Sinos, São Leopoldo, 2013.

NAIR, V.; HINTON, G. E. Rectified Linear Units Improve Restricted Boltzmann Machines. In: ICML, 2010. Anais... Omnipress, 2010. p. 807–814.

NOROUZI, M. Convolutional Restricted Boltzmann Machines for Feature Learning. 2009. Dissertação (Mestrado em Ciência da Computação) — Simon Fraser University, Canada, 2009.

PALTOGLOU, G.; THELWALL, M. A Study of Information Retrieval Weighting Schemes for Sentiment Analysis. In: ANNUAL MEETING OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS, 48., 2010, Stroudsburg, PA, USA. Proceedings... Association for Computational Linguistics, 2010. p. 1386–1395. (ACL '10).

PAN, S. J.; NI, X.; SUN, J.-T.; YANG, Q.; CHEN, Z. Cross-domain Sentiment Classification via Spectral Feature Alignment. In: INTERNATIONAL CONFERENCE ON WORLD WIDE WEB, 19., 2010, New York, NY, USA. Proceedings... ACM, 2010. p. 751–760. (WWW '10).

PANG, B.; LEE, L. A Sentimental Education: sentiment analysis using subjectivity summarization based on minimum cuts. In: ND ANNUAL MEETING ON ASSOCIATION FOR COMPUTATIONAL LINGUISTICS, 42., 2004, Stroudsburg, PA, USA. Proceedings... Association for Computational Linguistics, 2004.

PANG, B.; LEE, L. Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval, [S.l.], v. 2, n. 1-2, p. 1–135, 2008.

PORTER, M. F. Readings in Information Retrieval. In: . San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1997. p. 313–316.

READ, J.; PFAHRINGER, B.; HOLMES, G.; FRANK, E. Classifier Chains for Multi-label Classification. In: EUROPEAN CONFERENCE ON MACHINE LEARNING AND KNOWLEDGE DISCOVERY IN DATABASES: PART II, 2009, Berlin, Heidelberg. Proceedings... Springer-Verlag, 2009. p. 254–269. (ECML PKDD '09).

SALAKHUTDINOV, R. Learning Deep Generative Models. 2009. Tese (Doutorado em Ciência da Computação) — University of Toronto, Canada, 2009.

SALAKHUTDINOV, R.; HINTON, G. Semantic Hashing. Int. J. Approx. Reasoning, New York, NY, USA, v. 50, n. 7, p. 969–978, jul 2009a.

SALAKHUTDINOV, R.; HINTON, G. Deep Boltzmann Machines. In: INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE AND STATISTICS, 2009b. Proceedings... [S.l.: s.n.], 2009b. v. 5, p. 448–455.

SALTON, G.; MCGILL, M. J. Introduction to Modern Information Retrieval. New York, NY, USA: McGraw-Hill, Inc., 1986.

SANTOS, M. A. M. R. dos. Extraindo Regras de Associação a partir de Textos. 2002. Dissertação de Mestrado — Programa de Pós-Graduação em Informática Aplicada - Pontifícia Universidade Católica do Paraná, Curitiba, 2002.

SOCHER, R. Recursive Deep Learning for Natural Language Processing and Computer Vision. 2014. Tese (Doutorado em Ciência da Computação) — Stanford University, California, 2014.

SOCHER, R.; MANNING, C. Deep Learning for NLP (without Magic). In: HLT-NAACL, 2013. Anais... The Association for Computational Linguistics, 2013. p. 1–3.

TANG, Y. Robust Boltzmann Machines for Recognition and Denoising. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR), 2012., 2012, Washington, DC, USA. Proceedings... IEEE Computer Society, 2012. p. 2264–2271. (CVPR '12).

TANYILDIZI, H. Fuzzy Logic Model for the Prediction of Bond Strength of High-strength Lightweight Concrete. Adv. Eng. Softw., Oxford, UK, v. 40, n. 3, p. 161–169, Mar. 2009.

VICO, D. D. Deep Neural Networks. 2012. Dissertação (Mestrado em Ciência da Computação) — Universidad Autónoma de Madrid, Espanha, 2012.

VINCENT, P.; LAROCHELLE, H.; BENGIO, Y.; MANZAGOL, P.-A. Extracting and Composing Robust Features with Denoising Autoencoders. In: INTERNATIONAL CONFERENCE ON MACHINE LEARNING, 25., 2008, New York, NY, USA. Proceedings... ACM, 2008. p. 1096–1103. (ICML '08).

YANG, Y. Learning Hierarchical Representations for Video Analysis Using Deep Learning. 2013. Tese (Doutorado em Ciência da Computação) — University of Central Florida, Florida, 2013.

YESSENALINA, A.; YUE, Y.; CARDIE, C. Multi-level Structured Models for Document-level Sentiment Classification. In: CONFERENCE ON EMPIRICAL METHODS IN NATURAL LANGUAGE PROCESSING, 2010., 2010, Stroudsburg, PA, USA. Proceedings... Association for Computational Linguistics, 2010. p. 1046–1056. (EMNLP '10).

YU, D.; DENG, L.; WANG, S. Learning in the Deep-Structured Conditional Random Fields. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS, 2009. Anais... NIPS 2009 Workshop on Deep Learning for Speech Recognition and Related Applications, 2009.

ZHOU, S.; CHEN, Q.; WANG, X. Active Deep Networks for Semi-supervised Sentiment Classification. In: INTERNATIONAL CONFERENCE ON COMPUTATIONAL LINGUISTICS: POSTERS, 23., 2010, Stroudsburg, PA, USA. Proceedings... [S.l.: s.n.], 2010. p. 1515–1523.

ZHOU, S.; CHEN, Q.; WANG, X. Active deep learning method for semi-supervised sentiment classification. Neurocomputing, [S.l.], v. 120, p. 536–546, 2013.

ZHOU, S.; CHEN, Q.; WANG, X. Fuzzy deep belief networks for semi-supervised sentiment classification. Neurocomputing, [S.l.], v. 131, n. 0, p. 312 – 322, 2014.

APÊNDICE A *ALGORITMO CONTRASTIVE DIVERGENCE*

AlgoritmoCD

RBMUpdate(x1, ϱ , *W*, b, c)

Algoritmo de atualização de unidades de RBM. x1 é uma amostra de treinamento para o RBM ϱ é a taxa de aprendizado

W é a matriz com os pesos sinápticos das conexões entre as unidades visíveis e ocultas. b é o vetor de *bias* das unidades visíveis

c é o vetor de *bias* das unidades ocultas

for all unidades ocultas *i do*

compute $Q(h_{1i} = 1|x_1)$ (para unidades binomiais, $sigm(c_i + \sum_j W_{ij}x_{1j})$)

sample $h_{1i} \in 0, 1$ de $Q(h_{1i} = 1|x_1)$

end for

for all unidades visíveis *j do*

compute $P(h_{2j} = 1|h_1)$ (para unidades binomiais, $sigm(b_j + \sum_i W_{ij}h_{1i})$)

sample $x_{2j} \in 0, 1$ de $P(h_{2j} = 1|h_1)$

end for

for all unidades ocultas *i do*

compute $Q(h_{2i} = 1|x_2)$ (para unidades binomiais, $sigm(c_i + \sum_j W_{ij}x_{2j})$)

end for

$W \leftarrow W + \varrho\,(h_1 x_1' - Q(h_2 = 1\,|x_2))x_2')$

$b \leftarrow b + \varrho\,(x_1 - x_2)$

$c \leftarrow c + \varrho\,(h_1 - Q(h_2 = 1\,|x_2))$

APÊNDICE B VALORES DE PRECISÃO

Tabela 21: Precisão (%) POS - GPS - 120 épocas *Pre-training* e *Fine-tuning*

Camadas	Número de Termos						
1ª, 2ª, 3ª	300	500	1000	2000	3000	4000	5000
100, 100, 200	66,5	65,0	68,0	74,0	74,7	74,7	74,9
200, 200, 400	64,6	65,2	64,4	64,7	65,0	64,8	65,6
300, 300, 600	65,8	66,1	65,2	64,8	64,9	64,1	64,4
400, 400, 800	64,9	63,8	63,8	64,7	64,5	64,7	64,6
500, 500, 1000	62,8	63,9	63,7	64,6	64,7	64,4	64,9

Tabela 22: Precisão (%) NEG - GPS - 120 épocas *Pre-training* e *Fine-tuning*

Camadas	Número de Termos						
1ª, 2ª, 3ª	300	500	1000	2000	3000	4000	5000
100, 100, 200	75,8	77,2	75,1	70,4	72,2	72,6	72,1
200, 200, 400	77,6	78,0	77,3	78,0	77,4	77,1	75,9
300, 300, 600	78,2	77,4	78,1	80,4	78,5	80,1	79,8
400, 400, 800	78,4	80,7	78,9	79,7	80,2	78,2	79,4
500, 500, 1000	81,3	80,3	81,0	80,1	79,1	79,3	80,1

Tabela 23: Precisão (%) POS - GPS - 160 épocas *Pre-training* e *Fine-tuning*

Camadas	Número de Termos						
1ª, 2ª, 3ª	300	500	1000	2000	3000	4000	5000
100, 100, 200	66,6	65,6	68,1	75,3	75,4	72,9	75,2
200, 200, 400	64,9	65,3	65,9	64,0	64,6	63,7	64,0
300, 300, 600	65,5	64,8	65,7	65,1	64,1	64,7	64,9
400, 400, 800	65,0	64,7	64,7	66,0	64,8	65,1	64,9
500, 500, 1000	62,9	63,0	62,6	63,4	62,9	63,9	62,9

Tabela 24: Precisão (%) NEG - GPS - 160 épocas *Pre-training* e *Fine-tuning*

Camadas	Número de Termos						
1ª, 2ª, 3ª	300	500	1000	2000	3000	4000	5000
100, 100, 200	75,2	77,4	73,1	71,9	71,4	72,2	72,2
200, 200, 400	77,9	78,0	76,8	77,8	76,7	78,6	78,4
300, 300, 600	77,6	78,3	77,4	77,0	79,2	78,7	78,3
400, 400, 800	76,1	78,3	79,9	76,4	79,3	79,7	79,8
500, 500, 1000	81,8	79,8	81,0	80,0	80,7	79,4	80,0

Tabela 25: Precisão (%) POS - câmeras - 120 épocas *Pre-training* e *Fine-tuning*

Camadas	Número de Termos						
1ª, 2ª, 3ª	300	500	1000	2000	3000	4000	5000
100, 100, 200	83,1	82,7	83,1	83,3	83,5	83,8	83,6
200, 200, 400	83,1	83,0	82,5	82,7	82,5	83,1	83,4
300, 300, 600	82,3	82,2	82,9	82,4	82,1	82,2	82,4
400, 400, 800	82,2	81,6	81,7	81,9	81,4	81,6	81,9
500, 500, 1000	81,0	81,2	81,4	81,5	81,8	81,3	81,4

Tabela 26: Precisão (%) NEG - câmeras - 120 épocas *Pre-training* e *Fine-tuning*

Camadas	Número de Termos						
1ª, 2ª, 3ª	300	500	1000	2000	3000	4000	5000
100, 100, 200	86,5	86,8	86,9	85,6	85,7	85,8	85,7
200, 200, 400	86,6	86,7	86,7	86,8	87,1	86,9	86,7
300, 300, 600	86,8	86,6	86,8	86,3	86,4	86,6	86,2
400, 400, 800	86,7	87,1	86,2	86,4	86,5	86,5	86,7
500, 500, 1000	86,5	86,6	86,5	86,5	86,6	86,4	86,4

Tabela 27: Precisão (%) POS - câmeras - 160 épocas *Pre-training* e *Fine-tuning*

Camadas	Número de Termos						
1ª, 2ª, 3ª	300	500	1000	2000	3000	4000	5000
100, 100, 200	83,0	82,0	83,1	82,8	83,7	83,0	83,3
200, 200, 400	82,6	82,6	82,5	82,3	82,1	82,7	83,0
300, 300, 600	82,9	82,5	82,5	82,8	82,4	82,8	82,5
400, 400, 800	82,5	82,5	82,3	82,3	82,3	82,2	82,3
500, 500, 1000	81,5	81,8	82,1	81,9	80,9	81,7	81,9

Tabela 28: Precisão (%) NEG - câmeras - 160 épocas *Pre-training* e *Fine-tuning*

1ª, 2ª, 3ª	Número de Termos						
Configuração	300	500	1000	2000	3000	4000	5000
100, 100, 200	85,7	86,5	86,5	86,2	86,4	86,0	85,8
200, 200, 400	86,2	86,8	86,8	87,0	87,3	86,8	87,1
300, 300, 600	86,3	86,3	86,2	86,6	86,6	86,8	86,7
400, 400, 800	86,9	86,7	87,1	86,5	86,7	86,5	86,8
500, 500, 1000	86,6	86,9	87,3	87,0	87,2	87,1	86,4

Tabela 29: Precisão (%) POS - livros - 120 épocas *Pre-training* e *Fine-tuning*

Camadas	Número de Termos						
1ª, 2ª, 3ª	300	500	1000	2000	3000	4000	5000
100, 100, 200	74,8	75,1	74,8	73,7	72,5	72,8	71,2
200, 200, 400	74,6	75,0	74,4	73,4	73,1	72,2	72,1
300, 300, 600	73,3	72,7	73,4	73,4	73,1	72,7	72,6
400, 400, 800	72,8	72,6	73,0	72,5	72,6	72,4	71,8
500, 500, 1000	72,1	72,4	72,8	73,2	72,3	71,6	71,7

Tabela 30: Precisão (%) NEG - livros - 120 épocas *Pre-training* e *Fine-tuning*

Camadas	Número de Termos						
1ª, 2ª, 3ª	300	500	1000	2000	3000	4000	5000
100, 100, 200	79,6	80,5	81,2	80,6	79,4	79,7	78,5
200, 200, 400	80,4	80,7	81,0	80,3	80,6	80,2	80,1
300, 300, 600	80,7	80,3	80,5	80,7	80,9	80,5	80,7
400, 400, 800	80,4	80,3	80,4	80,1	80,2	80,2	80,0
500, 500, 1000	80,1	80,1	80,3	80,6	80,2	80,2	79,9

Tabela 31: Precisão (%) POS - livros - 160 épocas *Pre-training* e *Fine-tuning*

Camadas	Número de Termos						
1ª, 2ª, 3ª	300	500	1000	2000	3000	4000	5000
100, 100, 200	74,9	75,0	74,0	72,7	72,9	71,7	72,3
200, 200, 400	75,0	75,1	73,8	73,2	73,2	72,4	72,1
300, 300, 600	74,3	74,0	73,8	73,3	72,6	73,2	72,5
400, 400, 800	73,0	73,7	73,5	73,5	73,6	72,6	72,7
500, 500, 1000	73,3	73,6	73,1	73,4	73,1	72,2	72,1

Tabela 32: Precisão (%) NEG - livros - 160 épocas *Pre-training* e *Fine-tuning*

Camadas	Número de Termos						
1ª, 2ª, 3ª	300	500	1000	2000	3000	4000	5000
100, 100, 200	79,7	80,4	80,4	80,1	80,0	79,5	79,5
200, 200, 400	80,0	81,0	80,5	80,6	80,5	80,2	80,5
300, 300, 600	80,8	80,3	80,2	80,7	80,1	80,8	80,4
400, 400, 800	79,9	80,7	80,2	80,8	80,6	80,5	80,5
500, 500, 1000	80,0	80,6	80,3	80,7	80,3	79,9	80,1

Tabela 33: Precisão (%) POS - filmes - 120 épocas *Pre-training* e *Fine-tuning*

Camadas	Número de Termos						
1ª, 2ª, 3ª	300	500	1000	2000	3000	4000	5000
100, 100, 200	66,5	65,0	68,0	74,0	74,7	74,7	74,9
200, 200, 400	81,0	82,3	82,0	82,0	82,4	81,7	82,5
300, 300, 600	81,2	81,5	82,3	82,1	82,2	82,5	82,0
400, 400, 800	80,0	81,6	81,7	81,4	81,9	82,2	82,0
500, 500, 1000	77,4	80,6	81,9	82,3	81,9	82,2	81,8

Tabela 34: Precisão (%) NEG - filmes - 120 épocas *Pre-training* e *Fine-tuning*

Camadas	Número de Termos						
1ª, 2ª, 3ª	300	500	1000	2000	3000	4000	5000
100, 100, 200	75,8	77,2	75,1	70,4	72,2	72,6	72,1
200, 200, 400	77,2	71,1	70,3	70,0	70,0	69,8	69,8
300, 300, 600	80,6	80,9	81,4	81,4	81,5	81,1	81,0
400, 400, 800	79,5	80,3	80,9	81,1	80,6	81,3	80,7
500, 500, 1000	78,5	79,3	80,9	81,5	80,9	81,0	81,1

Tabela 35: Precisão (%) POS - filmes - 160 épocas *Pre-training* e *Fine-tuning*

Camadas	Número de Termos						
1ª, 2ª, 3ª	300	500	1000	2000	3000	4000	5000
100, 100, 200	82,2	83,4	83,3	83,4	83,4	83,2	83,3
200, 200, 400	81,8	82,0	82,8	81,9	82,2	82,0	81,8
300, 300, 600	81,4	82,2	82,7	83,0	82,7	82,6	83,0
400, 400, 800	81,1	82,1	82,5	82,8	82,7	82,6	82,7
500, 500, 1000	80,0	81,4	82,4	82,7	82,6	82,8	82,5

Tabela 36: Precisão (%) NEG - filmes - 160 épocas *Pre-training* e *Fine-tuning*

Camadas	Número de Termos						
1ª, 2ª, 3ª	300	500	1000	2000	3000	4000	5000
100, 100, 200	77,2	71,5	70,4	70,7	70,3	70,4	70,5
200, 200, 400	81,5	81,8	81,5	80,1	79,1	74,6	73,1
300, 300, 600	81,0	82,5	82,3	82,7	82,3	82,1	82,2
400, 400, 800	80,6	81,7	81,8	82,6	82,3	81,8	82,4
500, 500, 1000	79,4	80,9	82,0	82,1	82,3	82,7	82,0

APÊNDICE C VALORES DE *F-MEASURE*

Tabela 37: *F-measure* (%) - GPS - 120 épocas *Pre-training* e *Fine-tuning*

Camadas	Número de Termos						
1ª, 2ª, 3ª	300	500	1000	2000	3000	4000	5000
100, 100, 200	69,0	68,2	69,1	71,7	73,1	73,4	73,1
200, 200, 400	67,2	68,7	67,6	67,7	68,0	67,7	67,3
300, 300, 600	68,8	68,8	68,5	69,1	68,3	68,2	67,9
400, 400, 800	68,1	67,3	66,9	67,9	67,9	67,7	67,9
500, 500, 1000	66,7	67,2	67,1	67,3	67,7	67,5	68,2

Tabela 38: *F-measure* (%) - GPS - 160 épocas *Pre-training* e *Fine-tuning*

Camadas	Número de Termos						
1ª, 2ª, 3ª	300	500	1000	2000	3000	4000	5000
100, 100, 200	69,0	68,5	68,8	73,0	72,9	72,0	73,2
200, 200, 400	68,1	68,5	68,5	67,3	67,5	67,2	67,4
300, 300, 600	68,4	67,9	68,3	67,9	67,7	68,2	68,2
400, 400, 800	67,1	67,7	68,0	68,3	67,8	68,3	67,8
500, 500, 1000	66,6	66,1	66,1	66,4	66,3	66,8	66,0

Tabela 39: *F-measure* (%) - câmeras - 120 épocas *Pre-training* e *Fine-tuning*

Camadas	Número de Termos						
1ª, 2ª, 3ª	300	500	1000	2000	3000	4000	5000
100, 100, 200	84,6	84,5	84,8	84,3	84,5	84,6	84,5
200, 200, 400	84,6	84,7	84,7	84,6	84,2	84,8	84,5
300, 300, 600	84,0	84,4	84,7	84,2	84,1	84,1	84,0
400, 400, 800	84,2	83,8	83,6	83,8	83,7	83,8	84,0
500, 500, 1000	83,2	83,6	83,7	83,9	83,8	83,8	83,7

Tabela 40: *F-measure* (%) - câmeras - 160 épocas *Pre-training* e *Fine-tuning*

Camadas	Número de Termos						
1ª, 2ª, 3ª	300	500	1000	2000	3000	4000	5000
100, 100, 200	84,2	84,0	84,6	84,2	84,9	84,3	84,4
200, 200, 400	84,2	84,4	84,4	84,4	84,4	84,5	84,8
300, 300, 600	84,4	84,2	84,1	84,4	84,2	84,5	84,3
400, 400, 800	84,4	84,3	84,4	84,1	84,3	84,1	84,3
500, 500, 1000	83,7	84,0	84,4	84,1	83,2	84,1	83,9

Tabela 41: *F-measure* (%) - livros - 120 épocas *Pre-training* e *Fine-tuning*

Camadas	Número de Termos						
1ª, 2ª, 3ª	300	500	1000	2000	3000	4000	5000
100, 100, 200	76,9	77,4	77,5	76,5	75,3	75,4	73,5
200, 200, 400	77,1	77,4	77,2	76,3	76,1	75,3	75,2
300, 300, 600	76,3	76,0	76,2	76,3	76,3	75,5	76,2
400, 400, 800	75,9	75,6	75,8	75,7	75,6	75,6	74,7
500, 500, 1000	75,2	75,4	75,8	76,3	75,3	75,2	74,6

Tabela 42: *F-measure* (%) - livros - 160 épocas *Pre-training* e *Fine-tuning*

Camadas	Número de Termos						
1ª, 2ª, 3ª	300	500	1000	2000	3000	4000	5000
100, 100, 200	77,0	77,3	76,7	75,6	75,8	74,8	75,2
200, 200, 400	77,2	77,7	76,6	76,2	76,2	75,5	75,4
300, 300, 600	77,0	76,6	76,4	76,4	75,6	76,3	75,7
400, 400, 800	75,9	76,6	76,3	76,5	76,5	75,7	75,8
500, 500, 1000	76,1	76,5	76,0	76,4	76,0	75,2	75,3

Tabela 43: *F-measure* (%) - filmes - 120 épocas *Pre-training* e *Fine-tuning*

Camadas	Número de Termos						
1ª, 2ª, 3ª	300	500	1000	2000	3000	4000	5000
100, 100, 200	78,6	74,9	74,3	74,1	74,2	74,0	74,2
200, 200, 400	81,2	81,3	82,0	80,3	78,7	77,1	76,3
300, 300, 600	80,8	81,1	81,7	81,5	81,6	82,2	81,3
400, 400, 800	78,7	80,5	81,2	81,1	81,0	81,7	81,1
500, 500, 1000	77,2	79,7	81,3	81,7	81,3	81,5	81,2

Tabela 44: *F-measure* (%) - filmes - 160 épocas *Pre-training* e *Fine-tuning*

Camadas	Número de Termos						
1ª, 2ª, 3ª	300	500	1000	2000	3000	4000	5000
100, 100, 200	78,9	75,5	74,8	75,1	74,8	74,8	74,9
200, 200, 400	81,5	81,9	82,0	80,8	80,3	77,1	76,0
300, 300, 600	81,1	82,2	82,4	82,8	82,4	82,3	82,6
400, 400, 800	80,7	81,8	82,1	82,6	82,4	82,1	82,5
500, 500, 1000	79,6	81,1	82,1	82,3	82,4	82,7	82,1

APÊNDICE D VALORES DE *RECALL*

Tabela 45: *Recall* (%) POS - GPS -120 épocas *Pre-training* e *Fine-tuning*

Camadas	Número de Termos						
1ª, 2ª, 3ª	300	500	1000	2000	3000	4000	5000
100, 100, 200	80,2	83,1	78,1	68,0	70,3	71,3	70,1
200, 200, 400	83,9	83,9	83,8	83,9	82,7	83,0	81,3
300, 300, 600	83,4	82,2	84,2	87,0	84,7	87,1	86,3
400, 400, 800	84,5	87,8	86,0	85,7	86,6	84,2	85,4
500, 500, 1000	88,9	86,7	87,3	86,1	85,0	85,2	86,3

Tabela 46: *Recall* (%) NEG - GPS - 120 épocas *Pre-training* e *Fine-tuning*

Camadas	Número de Termos						
1ª, 2ª, 3ª	300	500	1000	2000	3000	4000	5000
100, 100, 200	59,0	55,0	61,4	75,6	76,1	75,6	76,2
200, 200, 400	53,4	54,8	53,3	53,5	54,8	54,5	55,9
300, 300, 600	55,6	56,9	54,3	52,2	53,6	51,0	51,9
400, 400, 800	53,3	49,8	50,5	52,8	51,6	53,2	52,6
500, 500, 1000	47,2	50,3	49,8	51,9	52,8	52,0	52,3

Tabela 47: *Recall* (%) POS - GPS - 160 épocas *Pre-training* e *Fine-tuning*

Camadas	Número de Termos						
1ª, 2ª, 3ª	300	500	1000	2000	3000	4000	5000
100, 100, 200	79,4	82,5	76,3	69,9	69,0	71,7	70,3
200, 200, 400	83,9	83,7	81,5	84,3	82,5	85,9	85,0
300, 300, 600	83,0	84,3	83,2	82,8	86,1	84,7	84,5
400, 400, 800	81,5	84,7	86,6	80,9	84,9	85,3	86,1
500, 500, 1000	89,1	87,1	88,5	87,0	87,7	85,7	87,2

Tabela 48: *Recall* (%) NEG - GPS - 160 épocas *Pre-training* e *Fine-tuning*

Camadas	Número de Termos						
1ª, 2ª, 3ª	300	500	1000	2000	3000	4000	5000
100, 100, 200	59,6	56,1	62,4	76,4	77,0	71,7	70,3
200, 200, 400	54,1	55,0	57,0	52,3	54,2	50,8	51,9
300, 300, 600	55,4	53,6	55,3	54,8	51,4	53,5	53,7
400, 400, 800	54,7	52,8	51,7	57,2	52,9	53,5	52,0
500, 500, 1000	47,2	48,2	46,9	48,7	47,9	50,6	48,0

Tabela 49: *Recall* (%) POS - câmeras - 120 épocas *Pre-training* e *Fine-tuning*

Camadas	Número de Termos						
1ª, 2ª, 3ª	300	500	1000	2000	3000	4000	5000
100, 100, 200	87,1	87,5	87,6	86,0	86,0	86,2	86,0
200, 200, 400	82,0	82,0	81,3	81,5	81,1	82,1	82,4
300, 300, 600	87,5	87,3	87,4	87,0	87,2	87,3	86,8
400, 400, 800	87,5	88,0	87,0	87,2	87,4	87,3	87,6
500, 500, 1000	87,6	87,6	87,4	87,4	87,5	87,3	87,3

Tabela 50: *Recall* (%) NEG - câmeras - 120 épocas *Pre-training* e *Fine-tuning*

Camadas	Número de Termos						
1ª, 2ª, 3ª	300	500	1000	2000	3000	4000	5000
100, 100, 200	82,1	81,5	82,0	82,5	82,9	83,0	83,0
200, 200, 400	87,2	87,3	87,4	87,4	87,9	87,6	87,2
300, 300, 600	80,9	80,9	81,8	81,3	80,8	80,9	81,3
400, 400, 800	80,8	79,9	80,4	80,6	79,9	80,1	80,4
500, 500, 1000	79,2	79,5	79,8	80,1	80,4	79,9	79,9

Tabela 51: *Recall* (%) POS - câmeras - 160 épocas *Pre-training* e *Fine-tuning*

Camadas	Número de Termos						
1ª, 2ª, 3ª	300	500	1000	2000	3000	4000	5000
100, 100, 200	86,2	87,3	87,2	86,7	86,9	86,5	86,3
200, 200, 400	86,8	87,5	87,5	87,8	88,2	87,6	87,8
300, 300, 600	86,8	86,9	86,9	87,2	87,3	87,5	87,4
400, 400, 800	87,7	87,4	87,9	87,2	87,4	87,2	87,6
500, 500, 1000	87,5	87,8	88,1	87,9	88,3	88,1	87,2

Tabela 52: *Recall* (%) NEG - câmeras - 160 épocas *Pre-training* e *Fine-tuning*

Camadas	Número de Termos						
1ª, 2ª, 3ª	300	500	1000	2000	3000	4000	5000
100, 100, 200	82,2	80,7	82,1	81,8	82,9	82,1	82,5
200, 200, 400	81,6	81,4	81,3	81,0	80,6	81,5	81,8
300, 300, 600	81,9	81,5	81,4	81,7	81,2	81,6	81,3
400, 400, 800	81,2	81,3	81,0	81,0	81,1	81,0	81,0
500, 500, 1000	80,0	80,2	80,7	80,4	78,4	80,2	80,6

Tabela 53: *Recall* (%) POS - livros - 120 épocas *Pre-training* e *Fine-tuning*

Camadas	Número de Termos						
1ª, 2ª, 3ª	300	500	1000	2000	3000	4000	5000
100, 100, 200	81,3	82,4	83,3	83,1	82,2	82,5	82,1
200, 200, 400	82,4	82,7	83,3	82,8	83,2	83,2	83,1
300, 300, 600	83,4	83,1	83,1	83,2	83,6	83,4	83,6
400, 400, 800	83,2	83,1	83,1	82,9	83,1	83,1	83,1
500, 500, 1000	83,1	83,0	83,1	83,3	83,2	83,4	83,0

Tabela 54: *Recall* (%) NEG - livros - 120 épocas *Pre-training* e *Fine-tuning*

Camadas	Número de Termos						
1ª, 2ª, 3ª	300	500	1000	2000	3000	4000	5000
100, 100, 200	72,5	72,5	71,8	70,2	68,6	68,7	65,9
200, 200, 400	71,8	72,3	71,2	69,9	69,3	67,8	67,7
300, 300, 600	69,5	68,7	69,7	69,8	69,0	68,6	68,3
400, 400, 800	68,8	68,6	69,0	68,5	68,5	68,2	67,2
500, 500, 1000	67,7	68,2	68,9	69,3	68,0	66,9	67,1

Tabela 55: *Recall* (%) POS - livros - 160 épocas *Pre-training* e *Fine-tuning*

Camadas	Número de Termos						
1ª, 2ª, 3ª	300	500	1000	2000	3000	4000	5000
100, 100, 200	74,9	75,0	74,0	72,7	72,9	71,7	72,3
200, 200, 400	75,0	75,1	73,8	73,2	73,2	72,4	72,1
300, 300, 600	74,3	74,0	73,8	73,3	72,6	73,2	72,5
400, 400, 800	82,5	83,2	82,7	83,3	83,1	83,3	83,3
500, 500, 1000	82,4	83,1	83,0	83,4	82,9	82,9	83,1

Tabela 56: *Recall* (%) NEG - livros - 160 épocas *Pre-training* e *Fine-tuning*

Camadas	Número de Termos						
1ª, 2ª, 3ª	300	500	1000	2000	3000	4000	5000
100, 100, 200	79,7	80,4	80,4	80,1	80,0	79,5	79,5
200, 200, 400	80,0	81,0	80,5	80,6	80,5	80,2	80,5
300, 300, 600	80,8	80,3	80,2	80,7	80,1	80,8	80,4
400, 400, 800	69,4	70,2	70,1	69,8	70,1	68,5	68,6
500, 500, 1000	69,9	70,2	69,3	69,6	69,4	67,9	67,8

Tabela 57: *Recall* (%) POS - filmes - 120 épocas *Pre-training* e *Fine-tuning*

Camadas	Número de Termos						
1ª, 2ª, 3ª	300	500	1000	2000	3000	4000	5000
100, 100, 200	75,3	64,5	63,4	62,9	62,8	62,8	62,3
200, 200, 400	81,0	81,0	80,6	79,3	75,6	71,8	69,3
300, 300, 600	80,4	80,7	81,1	81,2	81,2	80,6	80,6
400, 400, 800	79,5	79,8	80,7	81,0	80,1	81,1	80,3
500, 500, 1000	79,2	78,7	80,5	81,2	80,7	80,6	80,8

Tabela 58: *Recall* (%) NEG - filmes - 120 épocas *Pre-training* e *Fine-tuning*

Camadas	Número de Termos						
1ª, 2ª, 3ª	300	500	1000	2000	3000	4000	5000
100, 100, 200	82,0	86,0	86,0	86,1	86,5	85,9	86,7
200, 200, 400	81,6	81,7	82,4	81,6	81,5	82,3	83,6
300, 300, 600	81,2	81,6	82,4	82,1	82,3	82,8	82,2
400, 400, 800	79,0	81,9	81,8	81,4	82,1	82,2	82,2
500, 500, 1000	75,2	80,9	82,0	82,4	81,9	82,4	81,8

Tabela 59: *Recall* (%) POS - filmes - 160 épocas *Pre-training* e *Fine-tuning*

Camadas	Número de Termos						
1ª, 2ª, 3ª	300	500	1000	2000	3000	4000	5000
100, 100, 200	74,5	64,8	63,1	63,6	63,0	63,2	63,3
200, 200, 400	81,3	81,7	81,1	79,3	77,6	70,3	67,9
300, 300, 600	81,0	82,6	82,2	82,7	82,2	82,0	82,0
400, 400, 800	80,4	81,5	81,5	82,6	82,2	81,6	82,4
500, 500, 1000	79,1	80,7	81,9	82,0	82,2	82,7	81,8

Tabela 60: *Recall* (%) NEG - filmes - 160 épocas *Pre-training* e *Fine-tuning*

Camadas	Número de Termos						
1ª, 2ª, 3ª	300	500	1000	2000	3000	4000	5000
100, 100, 200	83,6	87,0	87,3	87,3	87,4	87,2	87,2
200, 200, 400	81,8	82,0	83,0	82,4	83,2	84,4	84,7
300, 300, 600	81,3	81,8	82,7	82,9	82,6	82,6	83,2
400, 400, 800	81,1	82,0	82,6	82,7	82,6	82,6	82,6
500, 500, 1000	80,1	81,4	82,4	82,7	82,5	82,7	82,5

APÊNDICE E CÓDIGO-FONTE

Exemplos de chamadas da function *fct_dl*:

```
epocaspt = 30;
epocasft = 120;
fct_dl(300, epocaspt, epocasft, [100 100 200]);
fct_dl(500, epocaspt, epocasft, [100 100 200]);

fct_dl(1000, epocaspt, epocasft, [100 100 200]);
```

Implementação *fct_dl*

```
function fct_dl(termos, maxepochpre, maxepochft, camadas)
    close all
    maxepoch =maxepochpre;
    numFolds = 10;
    qtamostrasteste = 100 ;
    totalacuracia=0;
    totalprecisao=0;
    totalrecall=0;
    qtcamadas = size(camadas(1));
    date=clock;
    camada1 = camadas(1); camada2 = camadas(2); camada3 = camadas(3);
    disp(clock);
    for i=1:numFolds,
        eval(['load wIG_fold_' num2str(i) '_' num2str(termos) 'T']);
        fprintf(1,'fold
        teste = tdmTestIgFreq;
        treino = tdmTrainIgFreq;
        n = size(treino); n = n(2); n = n/2;
        treino_nnx = treino';
        treino_nny = tn';
        teste_nnx = teste';
        teste_nny = tes_tn';
        treino_pos = treino_nnx(1:n,:);
        treino_neg = treino_nnx((n+1):(n*2),:);
        teste_pos = teste_nnx(1:qtamostrasteste,:);
        teste_neg = teste_nnx(101:100+qtamostrasteste,:);
        treino_target_pos = treino_nny(1:n,:);
        treino_target_neg = treino_nny((n+1):(n*2),:);
        teste_target_pos = teste_nny(1:qtamostrasteste,:);
        teste_target_neg = teste_nny(101:100+qtamostrasteste,:);
        numhid=camada1; numpen=camada2; numpen2=camada3; numpen3 = 1000;
        makebatches;
        [numcases numdims numbatches]=size(batchdata);
        fprintf(1,'Pretraining Layer 1 with RBM: restart=1;
        rbm; %chamada da function de treinamento de RBM
        hidrecbiases=hidbiases;
        save mnistvhclassify vishid hidrecbiases visbiases;
        fprintf(1,'Layer 2 with RBM:
        batchdata=batchposhidprobs;
        numhid=numpen;
        restart=1;
        rbm; %chamada da function de treinamento de RBM
        hidpen=vishid; penrecbiases=hidbiases; hidgenbiases=visbiases;
        save mnisthpclassify hidpen penrecbiases hidgenbiases;
        fprintf(1,'Layer 3 with RBM:
        batchdata=batchposhidprobs;
        numhid=numpen2; restart=1;
        rbm; %chamada da function de treinamento de RBM
        hidpen2=vishid; penrecbiases2=hidbiases; hidgenbiases2=visbiases;
        save mnisthp2classify hidpen2 penrecbiases2 hidgenbiases2;
        maxepoch =maxepochft;
        backpropclassify; % chamada da function de fine-tuning
        An = compet(targetout(:,:)')';
        TN=0;FP=0;TP=0;FN=0;
        for y=1:size(target,1)
            if (target(y,:)==([1 0]))
                if (An(y,1)==1  An(y,2)==0)
                    TN = TN+1;
                else
                    FP=FP+1;
                end;
            else
                if (An(y,1)==0  An(y,2)==1)
                    TP = TP+1;
                else
```

```
                    FN=FN+1;
                end;
            end;                end;
        accuracy=(TP+TN)/(TP+FP+FN+TN);
        precisao = TP/(TP + FP);
        recall = TP/(TP + FN);
        totalacuracia=totalacuracia+accuracy;
        totalprecisao=totalprecisao+precisao;
        totalrecall=totalrecall+recall;
        best_ac(i)=accuracy;
        recall_neg=TN/(TN+FP);
        recall_pos=TP/(TP+FN);
        precision_neg=TN/(TN+FN);
        precision_pos=TP/(TP+FP);
        fmeasure_neg=2*recall_neg*precision_neg/(recall_neg+precision_neg);
        fmeasure_pos=2*recall_pos*precision_pos/(recall_pos+precision_pos);
        best_rec_pos(i)= recall_pos;
        best_rec_neg(i)= recall_neg;
        best_prec_pos(i) = precision_pos;
        best_prec_neg(i) = precision_neg;
        best_fmeans_pos(i) = fmeasure_pos;
        best_fmeans_neg(i) = fmeasure_neg;
        proporcao_neg = sum(tes_tn(1,:)==1)/size(tes_tn,2);
        proprocao_pos = sum(tes_tn(2,:)==1)/size(tes_tn,2);
        overall_recall=     (proporcao_neg*recall_neg)+(proprocao_pos*recall_pos);
        overall_precision=      (proporcao_neg*precision_neg)+(proprocao_pos*precision_pos);
        overall_fmeasure=       (proporcao_neg*fmeasure_neg)+(proprocao_pos*fmeasure_pos);
        oR=overall_recall;
        oP=overall_precision;
        oFm=overall_fmeasure;
        best_oR(i)=oR;
        best_oP(i)=oP;
        best_oFm(i)=oFm;
        num2str(FN)]);
        disp([’precisao e recal: ’ num2str(precisao) ’ ’ num2str(recall)]);
end;
```

```
    disp([’Média acuracia fold: ’ num2str(totalacuracia/numFolds)]);
    disp([’Média recall fold: ’ num2str(totalrecall/numFolds)]);
    disp([’Média precisao fold: ’ num2str(totalprecisao/numFolds)]);

end
```