Um Modelo de Rede Neural Convolucional para Classificação de Peças de Vestuário

Anita Maria da Rocha Fernandes
Ciência da Computação
Universidade do Vale do Itajaí - UNIVALI
São José, SC, Brasil
anita.fernandes@univali.com

Andrei Hodecker
Pós Graduação em Big Data
Universidade do Vale do Itajaí - UNIVALI
São José, SC, Brasil
andreih@outlook.com

ABSTRACT

An algorithm capable of identifying clothing parts can be very useful for identifying a person´s social identity, among other applications. Convolutional neural networks models have been shown to be efficient in the task of image classification. This paper explores and analyzes models of convolutional neural networks in the task of classifying clothing parts by images. The models presented in this paper showed higher accuracy compared to non-convolutional models of the literature.

KEYWORDS

Convolutional neural networks, Clothing parts identification, Image analysis.

1 Introdução

Algoritmos capazes de identificar peças de vestuários podem ser usados para aprender o estilo de moda que o comprador procura e oferecer recomendações de produtos em lojas virtuais [1], porém todos estes produtos precisam estar previamente anotados e categorizados, e apenas uma porcentagem muito pequena das fotografias na internet possuem metadados relacionados a moda, isto indica que um algoritmo capaz de extrair informações de imagens pode ser mais interessante que anotações manuais submetidas a humanos [2].

As aplicações práticas para um algoritmo de análise visual de vestuário são inúmeras. Brossard et al [3] apresentam um método de aprendizado multi-classe, utilizando *Random Forest* e SVM (*Support Vector Machine*), capaz de identificar quinze classes, como por exemplo, vestido longo, jaqueta, casaco, entre outros. Wang et al [4], propõem além de um classificador, um detector de pontos de referência (*landmark*), utilizando um modelo de rede neural bidiresional convolucional recorrente.

Uma outra aplicação possível é a previsão de tendências de moda. Al-Halah et al. [5] apresentam um modelo não supervisionado com o objetivo de prever popularidades de estilos de moda, por exemplo, utilizando os dados de vendas de vestidos femininos em uma loja virtual, predizer os vestidos mais vendidos nos próximos 12 meses. Os resultados indicam que a análise visual é uma boa forma de fazer predição, superando dados textuais e metadados.

Kalantidis et al. [2] exploram a ideia de automaticamente sugerir produtos de vestuário usando uma única imagem como referência, em um primeiro estágio identificando as categorias de roupas na imagem e em seguida buscando peças similares. Já em Yamaguchi et al. [1], além de um categorizador de roupas, são apresentados resultados inicias promissores no uso de dados do vestuário para estimar poses. Um dos datasets mais populares com imagens de peças de vestuário é o Fashion-MNIST. Ele foi criado originalmente com o intuito de substituir por imagens de peças de vestuários o amplamente utilizado MNIST [6], que consiste em imagens dígitos números escritos à mão, fornecendo assim um conjunto de dados para classificação mais desafiador para benchmarks de técnicas de inteligência artificial [7].

Assim como MNIST, o dataset Fashion-MNIST consiste em 10 categorias para classificação e 70000 imagens em escala de cinza com 28 pixels de altura, por 28 pixels de largura. As imagens são extraídas de uma loja virtual de roupas, selecionando apenas a imagem frontal do produto, e em seguida processadas da seguinte forma: são convertidas para o formato PNG, redimensionadas para 28 pixels, o produto é centralizado e por fim convertido em escala de cinza [7].

Os produtos são categorizados e revisados por uma equipe de especialistas em moda da própria loja virtual, e as classes possíveis são: camisetas, calças, moletom, vestido, casaco, sandálias, camisas, sapatos), bolsas e botas [7].

Neste contexto, este trabalho apresenta uma pesquisa referente a exploração e análise do uso de técnicas de *deep learning* na tarefa de classificação de peças de vestuário utilizando o dataset Fashion-MNIST, e compara sua acuracidade em relação a outros modelos de aprendizagem de máquina já apresentados na academia, a fim de oferecer evidências para trabalhos futuros no contexto de classificação de imagens.

2 Desenvolvimento

Foram propostos quatro modelos de redes neurais convolucionais, inspiradas principalmente no trabalho de LeCun

et al. [6]. Todas utilizam a mesma inicialização de parâmetros para que seja justo a comparação entre elas. A inicialização de He [8] foi escolhida, pois esta demonstra superioridade em convergir com ativações não-lineares [8] A acurácia e perda (loss) da rede neural foram aferidas utilizando os datasets de treinamento e teste já divididos randomicamente em [7]. Isto é importante para que seja possível comparar estes modelos com outros da literatura que também fazem uso dessa metodologia de avaliação. Os modelos propostos foram nomeados com os seguintes rótulos: cnn-*dropout*-1, cnn-*dropout*-2, cnn-*droupout*-3 e cnn-*simple*-1. Todas fazem uso da função de ativação ReLU (*Rectified Linear Units*) nas camadas treináveis, com exceção da última camada, que utiliza softmax para classificar entre as 10 possíveis categorias.

Os modelos cnn-*dropout*-1 e cnn-*dropout*-3 fazem uso de dois blocos seguidos contendo: uma convolução, *max polling* e por fim *dropout*. Esses blocos, então, são ligados a mais duas camadas totalmente conectadas e por fim a camada de saída de 10 neurônios, cada um representando uma categoria. A única diferença entre os dois modelos é que o cnn-*dropout*-3 possui valores de *dropout* consideravelmente mais baixos.

Na Figura 1 é apresentada uma representação desse modelo, bem como os valores utilizados nos *kernels* (k), *filters* (f), *strides* (s) e *neurons* (n). Esta topologia contém cerca de 44.426 parâmetros treináveis.

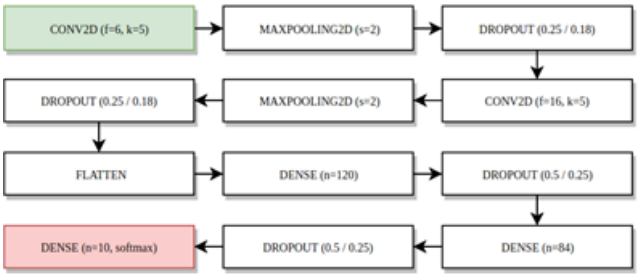

Figura 1. Topologia dos modelos cnn-dropout-1 e cnn-dropout-3.

O modelo cnn-*droupout*-2 foi proposto com um algumas camadas a mais que os outros, é muito similar ao modelo cnn-*dropout*-1, porém conta com duas camadas de convoluções antes da camada de *max pooling*. Além disso, ele possui uma convolução extra. Este modelo possui cerca de 32.340 parâmetros treináveis e é representado na Figura 2.

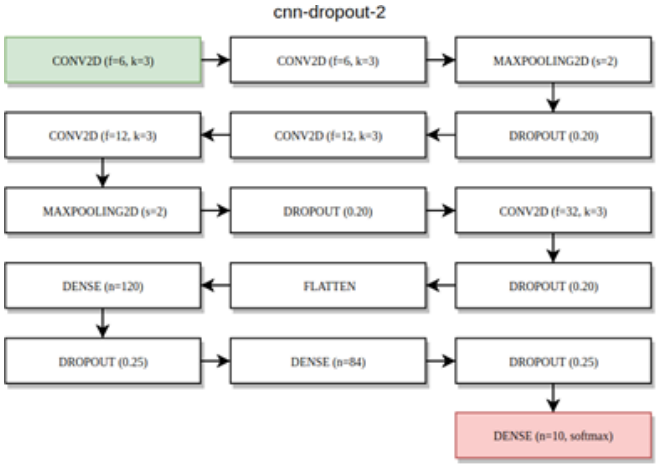

Figura 2. Topologia do modelo cnn-dropout-2.

Em contraste com os modelos apresentados, foi decidido também analisar um modelo com um menor número de camadas. O cnn-*simple*-1 apresenta apenas duas convoluções, seguidas de uma camada totalmente conectada, além dos respectivos *dropout* e *max pooling* semelhantes aos outros modelos. Este modelo possui 110.968 parâmetros treináveisv e é apresentado na Figura 3.

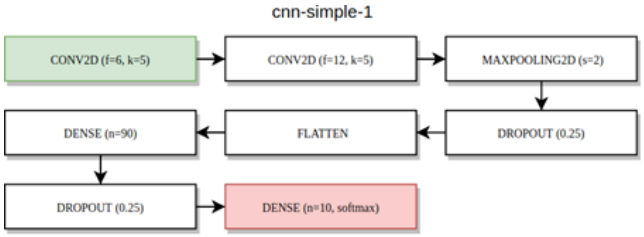

Figura 3. Topologia do modelo cnn-dropout-2.

Todos os modelos foram programados utilizando a linguagem de programação Python e o framework Keras [9], utilizando como *backend* a biblioteca Tensorflow [10]. As execuções dos treinamentos foram realizadas na plataforma Google Colab, pois esta fornece acesso a computação por GPU (*Graphics Processing Unit*), o que torna o processo mais rápido.

3 Considerações Finais

Foram avaliados os quatro modelos propostos neste artigo, todos treinados com 500 épocas e observando como convergiam em sua acurácia (porcentagem de acerto) e perda (o quão distante o modelo está do resultado esperado). Por fim, foi calculado tanta a acurácia no dataset de treinamento, quanto no de testes, para fins

de identificar um possível viés. A Tabela 1 apresenta os resultados obtidos para cada um dos modelos.

Tabela 1. Resultados obtidos dos modelos propostos

Modelo	Tempo	Loss (Trein./Teste)	Acurácia (Trein./Teste)
cnn-dropout-1	6m	0,21/0.26	91,87/90,35
cnn-dropout-2	9m 16s	0,19/0.25	92,59/90,81
cnn-dropout-3	5m 58s	0,14/0.25	94,53/90,86
cnn-simple-1	7m 52s	0,04/0.26	98,91/91,72

O modelo com maior acurácia foi o cnn-*simpl*e-1, com 91,72%, porém com auto viés, visto que atingiu 7,19% de diferença entre o dataset de treinamento e de testes. Já o modelo cnn-dropout-3 também apresentou viés, pois este possui dropout consideravelmente menor que seu semelhante cnn-*dropout*-1. O modelo com mais tempo de treinamento foi o cnn-*dropout*-2, com 9 minutos e 16 segundos, proveniente da quantidade maior de camada em relação a seus pares. Obteve expressivos 90,86% e manteve um viés relativamente baixo.

Utilizando o mesmo *dataset* de testes, é possível comparar estes modelos com algoritmos tradicionais não-convolutivos de aprendizagem de máquina. Em relação aos modelos avaliados em Xiao et al [7], todos os quatro modelos convolucionais deste artigo obtiveram acurácia superior ao modelo com maior acurácia, como pode-se observar na Tabela 2. Foi considerado na tabela apenas o melhor resultado de cada um destes algoritmos.

Tabela 2. Comparação com os modelos de Xiao et al [7]

Modelo	Acurácia (Teste)
cnn-simple-1	**91,72**
cnn-dropout-3	**90,86**
cnn-dropout-2	**90,81**
cnn-dropout-1	**90,35**
SVC	89,7
GradientBoostingClassifier	88,0
RandomForestClassifier	87,3
MLPClassifier	87,1
KNeighborsClassifier	85,4
LogisticRegression	84,2
LinearSVC	83,6
SGDClassifier	81,9
DecisionTreeClassifier	79,8
Perceptron	78,2
PassiveAggressiveClassifier	77,6
ExtraTreeClassifier	77,5

O modelo com maior acurácia foi o cnn-*simpl*e-1, com 91,72%, porém com auto viés, visto que atingiu 7,19% de diferença entre o dataset de treinamento e de testes. Já o modelo cnn-*dropout*-3 também apresentou viés, pois este possui *dropout* consideravelmente menor que seu semelhante cnn-*dropout*-1. O modelo com mais tempo de treinamento foi o cnn-*dropout*-2, com 9 minutos e 16 segundos, proveniente da quantidade maior de camada em relação a seus pares. Obteve expressivos 90,86% e manteve um viés relativamente baixo.

Com os resultados evidenciados neste artigo, é possível concluir que modelos convolucionais possuem, de fato, mais acurácia na tarefa de classificação de peças de vestuário em relação a modelos de aprendizagem convencionais. Além disso, foi possível observar que a técnica dropout e mais camadas convolutivas são eficazes em diminuir o viés de um determinado modelo. Contudo, não foram exploradas neste artigo técnicas de augmentation no dataset de imagens, esta técnica deve diminuir consideravelmente o viés e aumentar a capacidade de generalização dos modelos.

REFERÊNCIAS

[1] K. Yamaguchi, K, M. H. Kiapour, L. E. Ortiz, T. L. Berg. (2012). Parsing clothing in fashion photographs. In 2012 IEEE Conference on Computer Vision and Pattern Recognition, pages 3570–3577, Providence, RI. IEEE.

[2] Y. Kalantidis, L. Kennedy, L. J. Li. (2013). Getting the look: Clothing recognition and segmentation for automatic product suggestions in everyday photos. In Proceedings of the 3rd ACM Conference on International Conference on Multimedia Retrieval - ICMR '13, page 105, Dallas, Texas, USA. ACM Press.

[3] L. Bossard, M. Dantone, C. Leistner, C. Wengert, T. Quack, L. Van Gool (2013). Apparel Classification with Style. In D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J.C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Vardi, Y. Weikum, K. M. Lee, Y. Matsushita, J. M. Rehg, Z. Hu, editors, Computer Vision – ACCV 2012, volume 7727, pages 321–335. Springer Berlin Heidelberg.

[4] W. Wang, Y. Xu, J. Shen, S.C. Zhu. (2018). Attentive Fashion Grammar Network for Fashion Landmark Detection and Clothing Category Classification. page 10.

[5] Z. Al-Halah, R. Stiefelhagen, K. Grauman. (2017). Fashion Forward: Forecasting Visual Style in Fashion. In 2017 IEEE International Conference on Computer Vision (ICCV), pages 388–397, Venice. IEEE.

[6] Y. Le Cun, Y. Bengio, G. Hinton. (2015). Deep learning. Nature, 521(7553):436–444.

[7] H. Xiao, K. Rasul, R. Vollgraf. (2017). Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms. arXiv:1708.07747.

[8] K. He, X. Zhang, S. Ren, J. Sun. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. CoRR, abs/1502.01852.

[9] F. Chollet. (2015). Keras. https://keras.io.

[10] M. Abadi, A. Agarwal, P, Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane´, R. Monga, S., Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software disponível em tensorflow.org.