Understanding fully-connected and convolutional layers in unsupervised learning using face images

Lucas Fontes Buzuti
Department of Electrical Engineering
FEI University Center
São Bernardo do Campo-SP, Brazil
lucas.buzuti@outlook.com

Carlos Eduardo Thomaz
Department of Electrical Engineering
FEI University Center
São Bernardo do Campo-SP, Brazil
cet@fei.edu.br

***Abstract*—The goal of this paper is to implement and compare two unsupervised models of deep learning: Autoencoder and Convolutional Autoencoder. These neural network models have been trained to learn regularities in well-framed face images with different facial expressions. The Autoencoder's basic topology is addressed here, composed of encoding and decoding multilayers. This paper approaches these automatic codings using multivariate statistics to visually understand the bottleneck differences between the fully-connected and convolutional layers and the corresponding importance of the dropout strategy when applied in a model.**

***Index Terms*—deep neural network, autoencoder, convolutional autoencoder, multivariate statistics**

I. Introdução

Estudos relacionados a Inteligencia Artificial, com foco em Aprendizagem Profunda (Deep Learning) vêm mostrando resultados impressionantes na área de reconhecimento de padrões, chegando a superar o estado-da-arte [8] [11] [15]. No contexto de reconhecer padrões em imagens de faces, capturadas com diferentes expressões faciais, este artigo implementa e analisa um modelo de aprendizado não-supervisionado de redes neurais profundas denominado autoencoders, com o objetivo de compreender regularidades extraídas destas imagens.

Historicamente, em 1986 os autoencoders tiveram uma primeira citação indiretamente em um artigo relacionado ao erro de propagação [17], descrevendo um novo tipo de rede *feedforward* na época e seu formalismo matemático. A ideia ressurgiu em trabalhos subsequentes de pesquisa nos anos seguintes. Em 1989 Baldi e Hornik [2] introduziram os autoencoders propondo uma descrição precisa das características salientes da superfície anexada a função de erro quando as unidades são lineares. Durante as décadas de 1980 e 1990, diversos algoritmos de aprendizado não supervisionados que foram sugeridos para redes neurais puderam ser vistos como variações de dois métodos básicos: Análise de Componentes Principais (Principal Components Analysis ou PCA) e Quantização Vetorial (Vector Quantization ou VQ). Entretanto, em 1994, Hinton e Zemel [9] descreveram uma nova função objetiva para o treinamento de autoencoders que permitiu extrações de representações fatoriais não lineares. Usando o autoencoder para reduzir a dimensionalidade de dados, Hinton e Salakhutdinov comprovaram em 2006, a eficiência do autoencoder em relação ao PCA [8].

O uso de Deep Learning vem aumentando exponencialmente e esse crescimento está sendo possível porque redes neurais profundas com topologias muito complexas estão sendo computadas em GPUs (Graphics Processing Unit), permitindo comprovar na prática, o que no passado havia sido realizado matematicamente.

Neste artigo, será investigado um autoencoder profundo (Deep Autoencoder) [8] para análise de padrões em imagens faciais comparando dois tipos de camadas, sendo essas: Totalmente Conectada (Fully-Connected Layer) e Convolucional (Convolutional Layer). Tais redes neurais profundas contêm múltiplas camadas não lineares, tornando-as modelos adequados para aprender relações complexas entre entradas e saídas de imagens de faces, como exemplificado aqui.

II. Mapeamento de Entradas em Saídas

A. Autoencoders

Rumelhart, Hinton e Williams [17] descreveram um problema em que um conjunto de padrões de entradas são mapeados para um conjunto de padrões de saídas através de um número reduzido de neurônios/unidades ocultas (hidden units). Para provar o problema proposto, conjecturaram o mapeamento de N padrões binários de entrada para N padrões de saída, no qual N representa o número de unidades de entrada e saída, além disso presumiram o número de neurônios da camada oculta através de $\log_2 N$ [17].

O sistema proposto aprende a usar as unidades da camada oculta para formar um código com padrão binário distinto de cada N padrões de entrada. A topologia da rede realiza a codificação de N padrões de bits em $\log_2 N$ e então decodifica essas representações para os padrões de saída. O autoencoder pode ser descrito em duas partes: função do codificador (encoder function) e função do decodificador (decoder function), tal que a função

$$h = f(xW + b) \tag{1}$$

é a representação latente, x os dados de entrada, W matriz de pesos, b matriz bias e f a função de ativação, e

$$y = g(hW^T + c) \tag{2}$$

a entrada reconstruída, W^T matriz de pesos transposta, c matriz bias e g função de ativação. Os parâmetros do modelo são otimizados para minimizar o erro médio de reconstrução (average recontruction error), sendo descrito como:

$$\mathfrak{L}(x, y) = \theta^*, \theta^{'*} = \arg\min_{\theta,\theta'} \frac{1}{n} \sum_{i=1}^{n} L(x_i, y_i) \tag{3}$$

onde L é a função erro (loss function) dada pelo erro ao quadrado $L(x, y) =\| x - y \|^2$, $\theta = \{W, b\}$, $\theta^{'} = \{W^T, c\}$ e θ^*, $\theta^{'*}$ são os conjuntos de pesos e bias a serem atualizados e n é a quantidade de dados na entrada.

Autoencoders podem ser avaliados como sendo um caso especial de redes *feedforward* e podem ser treinados pelas técnicas típicas de gradiente descendente (gradient descent) e retro-propagação (back-propagation) [5] [8].

III. Dropout

Dropout é uma técnica de regularização que aborda dois tipos de soluções: evitar o overfitting e fornecer uma maneira de combinar (de forma exponencial) outras topologias de redes neurais diferentes de forma eficaz. O termo em redes neural "dropout" é uma palavra inglesa que tem como referência o abandono dos neurônios ocultos e/ou visíveis nos modelos. Ao referenciar o abandono dos neurônios ocultos ou neurônios visíveis, o intuito desse abandono é removê-los temporariamente da rede, juntamente com todas as suas conexões de entrada e saída. Essa remoção possibilita, na prática, o treinamento de "diversas topologias" em um único modelo sem a necessidade de modelar novas topologias separadamente e treiná-las. Também permite que o modelo não "memorize" um único dado especificamente [7].

A escolha de quais neurônios serão desativados é aleatória. Considerando a hipótese mais simples, cada neurônio é retido com uma probabilidade fixa p independente dos outros neurônios, tal que p pode ser escolhido usando uma análise de validação ou simplesmente ser definido em 0.5, que está próximo ao ideal em uma ampla gama de redes e tarefas [18]. Geralmente, para os neurônios de entrada, a probabilidade ótima de retenção é mais próxima de 1.0 do que 0.5 [18].

IV. Camada Convolucional

Camada convolucional é um dos conceitos mais utilizados de Deep Learning, permitindo inferir a representação da visão humana até o reconhecimento do que está sendo observado [12]. A praticabilidade dessa camada está na capacidade de compartilhamento de seus parâmetros com o espaço ortogonal, assim adquirindo uma representação tridimensional, tendo largura, altura e profundidade (feature maps). Essas camadas empilhadas são conhecidas como Redes Convolucionais (Convolutional Neural Networks ou ConvNets) [13] [19].

O cálculo da convolução se dá pelo deslizamento de uma distribuição (kernel) em cima de outra distribuição (e.g. $f * g$). Uma camada convolucional é composta por hiper-parâmetros para ser ajustada, tais como stride e padding [1]. Para fazer uma redução das distribuições com mais eficiência, geralmente são usados o Average Pooling ou Max Pooling, mas o pooling normalmente utilizado é o max [1].

Uma ConvNet, basicamente, é uma rede profunda em que ao invés de pilhas de camadas de multiplicações, tem-se pilhas de convoluções. A ideia nesses empilhamentos é se igualar a visão humana, com aplicação da convolução e pooling que progressivamente diminuirão as dimensões espaciais, enquanto aumentam a profundidade que corresponde à complexidade semântica de representação dos estímulos de entrada.

A. Autoencoder Convolucional

Autoencoders convencionais se utilizam de camadas totalmente conectadas, assim ignorando a estrutura natural de imagens 2D. Entretanto o uso de camadas convolucionais não é apenas uma solução para entradas de tamanhos realísticos, mas também introduz redundância nos parâmetros, forçando cada recurso a ser global, em outras palavras, abrange todo o campo visual. Os Autoencoders Convolucionais diferem dos convencionais, pois seus pesos são compartilhados, preservando a localidade espacial [4].

A topologia de um Autoencoder Convolucional é intuitivamente semelhante ao Autoencoder descrito na Seção II-A, exceto que os pesos são compartilhados [14]. Para uma entrada x mono-canal a representação latente do k-ésimo elemento do *feature map* [20] é dada por

$$h^k = f(x * W^k + b^k) \tag{4}$$

onde um único bias b por mapa de característica é usado e também transmitido para todo o mapa. A reconstrução é computada usando

$$y = g(\sum_{k \in H} h^k * \tilde{W}^k + c) \tag{5}$$

onde novamente há um bias c por canal de entrada. H identifica o grupo de *feature maps* latentes e $\tilde{W}$ identifica a operação de *flip* em ambas as dimensões dos pesos. A função de erro a ser minimizada será a mesma da Equação (3) e o treinamento pode ser pelas técnicas típicas de gradiente descendente e retro-propagação.

V. Experimentos e Resultados

Os experimentos foram desenvolvidos com o objetivo de abstrair o entendimento das camadas do Autoencoder (autoencoder convencional) e do Autoencoder Convolucional. Para a extração dos padrões da topologia proposta, utilizou-se uma base de dados disponível publicamente para pesquisa: FEI Face Database [3].

Da base de faces da FEI, foram utilizadas 400 imagens frontais contendo apenas o rosto, com expressão facial sorrindo e não sorrindo (normal), sendo 200 masculinas e 200 femininas.

A. *Processamento do Autoencoder*

Investigou-se a topologia do Autoencoder, ilustrado na Figura 1.

Foi utilizado dropout como regularizador [18] no codificador (encoder), no decodificador (decoder) e na camada de intersecção dos mesmos que geralmente é conhecida de bottleneck. Analisou-se a utilização do dropout como um artificio de modo a ter uma generalização no reconhecimento dos dados para o autoencoder convencional. A generalização evitará que os neurônios coadaptem no codificador, decodificador e principalmente no bottleneck, no qual os padrões dos dados estarão em uma dimensionalidade reduzida.

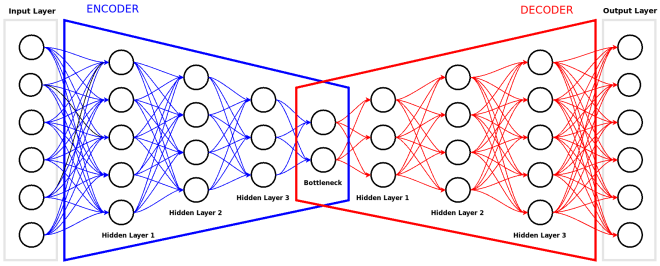

Fig. 1. Autoencoder proposto com camadas totalmente conectadas.

Foi aplicado à rede quatro tipos de probabilidade, sendo estas 1.0, 0.5, 0.1 e 0.01, em que quanto mais próximo de 1.0, mais neurônios estarão ativos, e mais próximo de zero mais desativados.

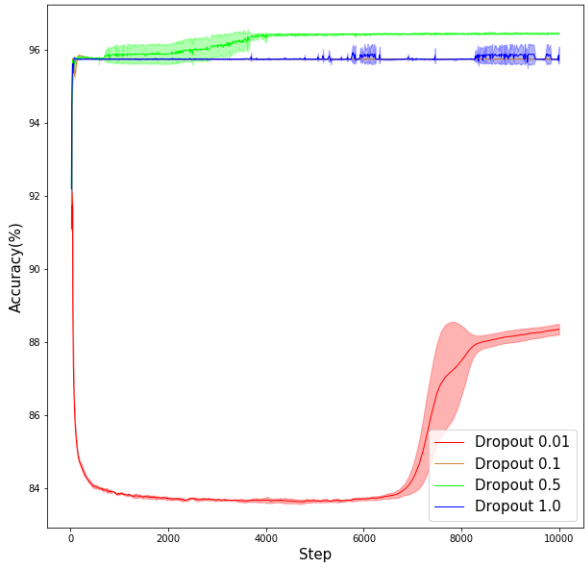

Fig. 2. Gráfico de teste da acurácia média em relação às quatro probabilidades (0.1, 0.5, 1.0) com 10000 épocas.

O uso de dropout mostra o quanto aumenta a eficiência no treinamento dessa rede neural [18]. Na Figura 2, ilustra-se a eficiência da acurácia média das quatro probabilidades. Na rede, na qual a probabilidade do dropout é 0.01, teve-se o pior resultado, pois 1% dos neurônios ficaram ativos durante o treinamento. Entretanto a probabilidade 0.5 visualmente é a melhor mesmo que no começo do treinamento tenha apresentado um desvio padrão expressivo em relação à média. Esse desvio padrão pode indicar que a rede não se estagnou em um mínimo local. Todavia, ao longo de seu treinamento, esse desvio ficou muito pequeno a ponto de ser considerado nulo. As duas probabilidades restantes praticamente apresentam o mesmo resultado, já que suas médias são análogas, sendo que o probabilidade 0.1 consta com um desvio padrão próximo de zero, assim não conseguindo se desenvolver em seu treinamento, podendo deduzir que o gradiente estacionou-se em um mínimo local. Quanto a probabilidade 1.0, tem-se a mesma analise da probabilidade 0.1, mas a diferença está que em parte do treinamento houve desvio padrão diferente de zero indicando a busca por um mínimo global.

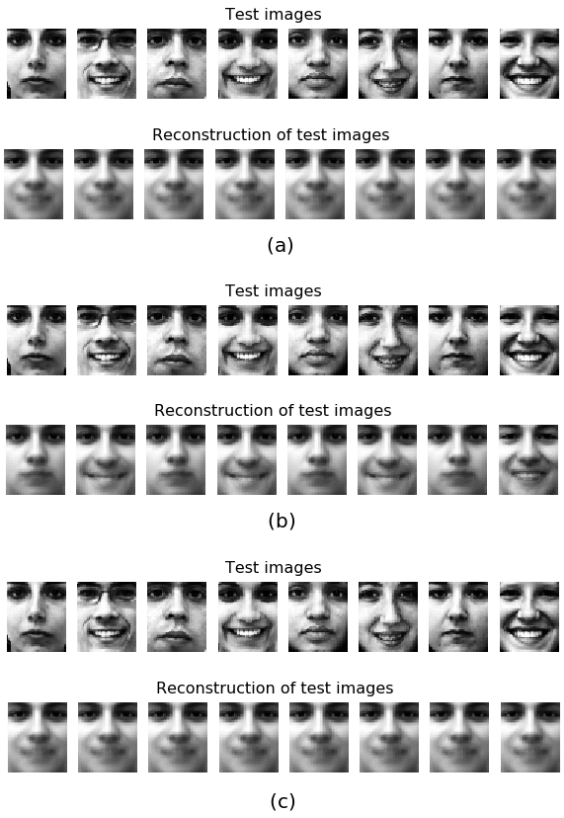

Fig. 3. Resultados do Autoencoder com as probabilidades 0.1 (a), 0.5 (b) e 1.0 (c).

As três melhores probabilidades (0.1, 0.5, 1.0) obtiveram uma acurácia relevante. Entretanto, visualizando o resultado (Figura 3) trazido pela topologia do Autoencoder proposto, conclui-se que a probabilidade 0.5 traz uma melhor generalização dos dados, pois além de reconstruir pontos semelhantes, tais como olhos, nariz e boca, a rede reproduz a expressão facial correspondente do dado original de entrada, como ilustrado na Figura 3.

Analisando visualmente os *feature maps* (mapas de características) da topologia com dropout 0.5, nota-se claramente que em ambas situações (face sorrindo e normal), há uma mescla nítida nas principais percepções, refletindo pixels de es-

tado sorrindo e normal, ilustrados na Figura 4. A combinação de cada peso contido no *feature map* representa o estado dos neurônios, os quais podem estar ativos, parcialmente ativos ou desativados, na iminência de definir os padrões dos dados de entrada.

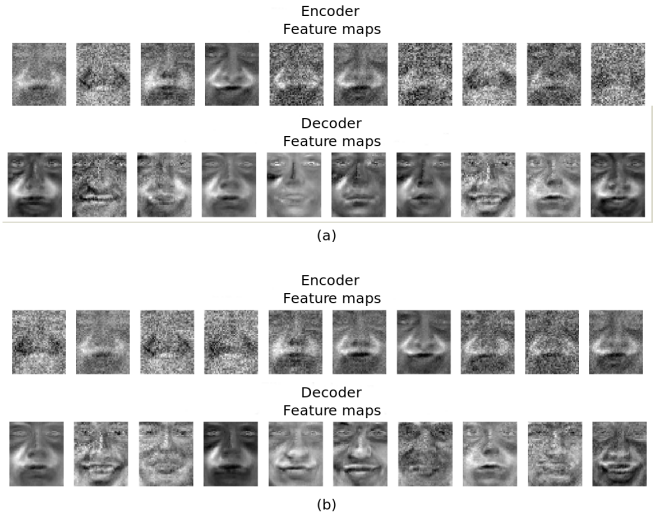

Fig. 4. Feature maps da primeira e última camadas do codificador e decodificador, respectivamente. O conjunto (a) representa os feature maps de uma face sorrindo e o conjunto (b) representa os feature maps de uma face normal.

Nos *feature maps*, os pixels cinzas estão representando os pesos zero e quanto mais branco mais positivos são os pesos (mais próximos de 1). Por outro lado, quanto mais escuros forem os pixels, mais negativos serão os pesos (mais próximos de -1). Os pixels positivos aumentarão a probabilidade de ativação dos neurônios, e pixels negativos diminuirão a probabilidade de um neurônio ser ativado.

Devido aos padrões das imagens serem compactados na camada de bottleneck e estarem em um espaço de dimensão muito grande, se torna inviável a sua visualização. Portanto, necessitou-se de uma redução de dimensionalidade, assim aplicando o PCA nos dados e obtendo as três primeiras componentes principais. A Figura 5 ilustra os dados no espaço R^3 com 98% de variância.

Pode-se verificar a esparsidade dos dados na Figura 5. Esse fenômeno pode ser explicado pela utilização de camadas totalmente conectadas, obrigando a vetorização dos dados de entrada. Assim, perde-se a relação espacial original dos pixels e a relação da vizinhança entre padrões.

B. Processamento do Autoencoder Convolucional

A topologia do Autoencoder Convolucional utilizada é análoga à topologia do Autoencoder descrito na Seção V-A. Contudo, o que diverge entre os Autoencoders é o tipo de camadas utilizadas. Para o processamento do Autoencoder Convolucional, camadas convolucionais são utilizadas e não há necessidade da utilização do dropout, pois camadas convolucionais tem em sua arquitetura conexões esparsas. Na Figura 6, descreve-se a topologia utilizada. Para comparar resultados,

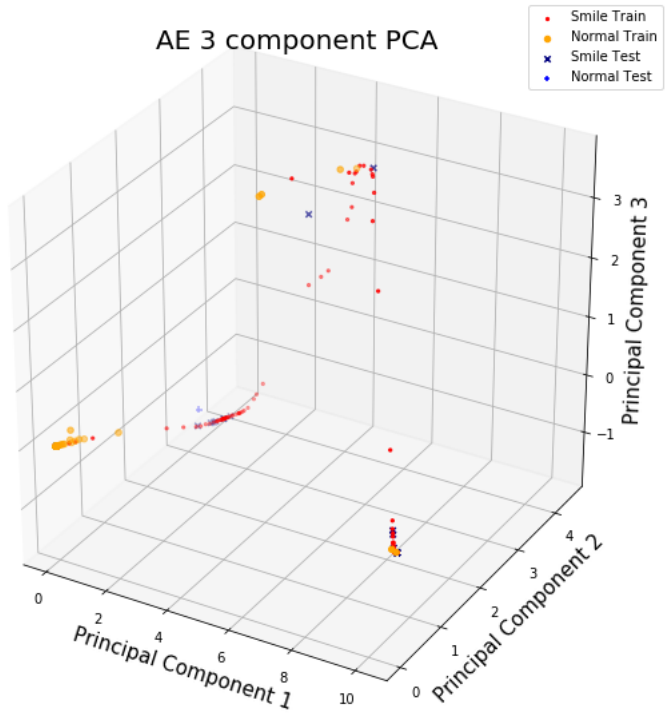

Fig. 5. Redução de dimensionalidade no bottleneck do Autoencoder.

entretanto, além de fazer o uso de camadas convolucionais foi empregado no bottleneck camadas totalmente conectadas, na mesma dimensão do bottleneck do autoencoder convencional.

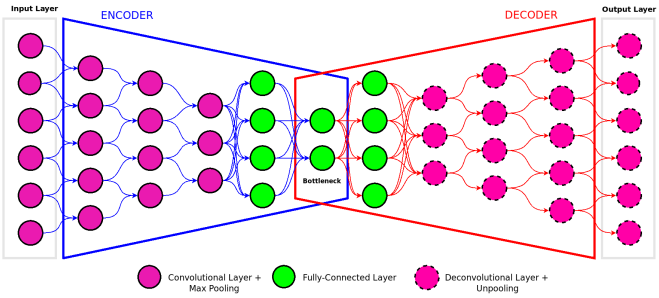

Fig. 6. Autoencoder Convolutional proposto.

Os resultados trazidos pela topologia convolucional demonstram que as relações entre padrões vizinhos foram preservadas, assim trazendo as reconstruções das baixas frequências de cada dado e a eliminação das altas frequências dos mesmos. Na Figura 7, pode-se visualizar claramente essas extrações e a eliminação das altas frequências na imagem em destaque.

De forma similar, analisando visualmente os *feature maps* da topologia convolucional, nota-se uma percepção distinta entre o estado sorrindo e normal, ilustrado na Figura 8. A combinação dos pesos resulta nos *feature maps* que representam atributos visuais de baixo e alto nível, na iminência de definir os padrões dos dados de entrada.

Atributos visuais de alto nível representados por conceitos semânticos inerentes às imagens de faces, como expressão

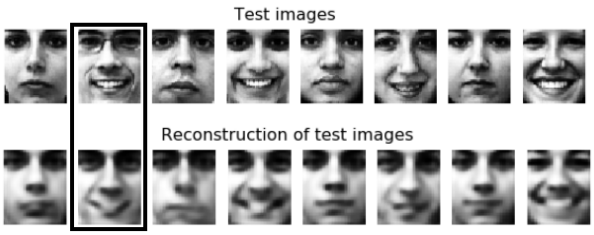

Fig. 7. Resultado do Autoencoder Convolucional com 1500 épocas de treinamento e 97.7% de acurácia de teste. A topologia foi capaz de reconstruir características expressivas e eliminar artefatos irrelevantes.

facial, e atributos visuais de baixo nível, como forma e textura, são detectados na primeira camada do codificador. Além dos atributos visuais inerentes às imagens de faces, o conceito de segmentação foi aprendido pela topologia, na qual a expressão facial foi segmentada. No decodificador, os *feature maps* resultaram nas baixas frequências da decodificação do bottleneck e atributos visuais de alto nível, então pode-se atrelar a eliminação das altas frequências na imagem em destaque na Figura 7 aos *feature maps* de baixas frequências.

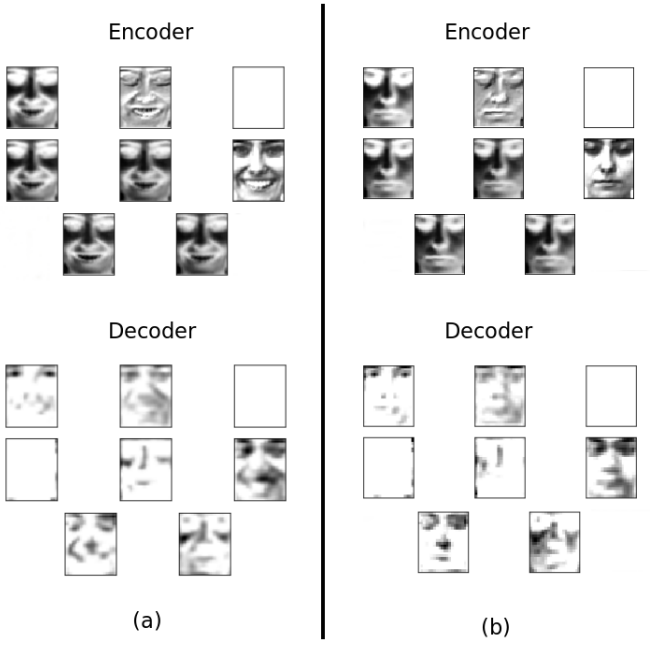

Fig. 8. Feature maps da primeira e última camadas do codificador e decodificador, sendo (a) feature maps de uma entrada de face sorrindo, e (b) feature maps de uma entrada de face normal.

Aplicando a mesma análise estatística no bottleneck, determinou-se as três primeiras componentes principais, assim passando os dados para um espaço de dimensão R^3 ilustrado na Figura 9. Neste caso, apenas aproximadamente 40% da variância total dos dados estão sendo representadas.

Nota-se, na Figura 9, um agrupamento melhor dos dados em relação à Figura 5, aponto de dividir o gráfico em duas classes: sorrindo e normal. Conjectura-se esse desempenho exclusivamente à utilização de camadas convolucionais. Usar apenas

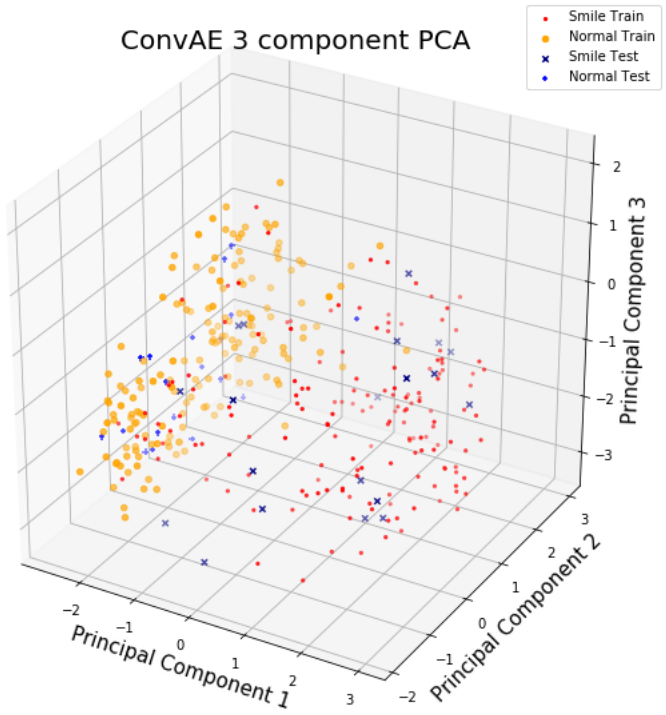

Fig. 9. Redução de dimensionalidade no bottleneck do Autoencoder Convolucional.

camadas totalmente conectadas, em que dados de natureza distintas sofram necessariamente vetorização, leva a perdas de informação de vizinhança entre os padrões aprendidos.

VI. Conclusão

Dropout é uma técnica de aprendizado profundo que provou a sua eficiência em redes neurais profundas, pois os principais problemas de overfitting e definição de topologia ótima conseguem ser corrigidos por essa técnica. Não há literatura sobre quantas camadas ocultas ou neurônios utilizar de forma determinística, por isso a importância do dropout no aprendizado profundo, principalmente nas topologias que fazem somente o uso de camadas totalmente conectadas.

Ao utilizar dados de natureza diferentes de unidimensionais, camadas convolucionais mostram-se eficientes, pois as relações de similaridade entre padrões vizinhos permanece. Todavia, camadas convolucionais são camadas de abstração de padrões, e há ainda a necessidade de utilizar camadas totalmente conectadas para a definição desses padrões.

Vislumbra-se, como trabalhos futuros, estender essas análises para outras topologias, como U-net [16], Variational Autoencoder [10] e Generative Adversarial Networks [6], e para outras bases de imagens de faces.

Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

REFERÊNCIAS

[1] ALBAWI, Saad; MOHAMMED, Tareq Abed; AL-ZAWI, Saad. Understanding of a convolutional neural network. In: IEEE. ENGINEERING and Technology (ICET), 2017 International Conference on. [S.l.: s.n.], 2017. p. 1–6.

[2] BALDI, Pierre; HORNIK, Kurt. Neural networks and principal component analysis: Learning from examples without local minima. Neural networks, Elsevier, v. 2, n. 1, p. 53–58, 1989.

[3] C. E. Thomaz and G. A. Giraldi. A new ranking method for Principal Components Analysis and its application to face image analysis, Image and Vision Computing, vol. 28, no. 6, pp. 902-913, June 2010.

[4] CHEN, Min et al. Deep features learning for medical image analysis with convolutional autoencoder neural network. IEEE Transactions on Big Data, 2017.

[5] GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. Deep Learning. [S.l.]: MIT Press, 2016. http://www.deeplearningbook.org.

[6] GOODFELLOW, Ian et al. Generative adversarial nets. In: ADVANCES in neural information processing systems. [S.l.: s.n.], 2014. p. 2672–2680.

[7] HINTON, Geoffrey E. et al. Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580, 2012.

[8] HINTON, Geoffrey E; SALAKHUTDINOV, Ruslan R. Reducing the dimensionality of data with neural networks. science, American Association for the Advancement of Science, v. 313, n. 5786, p. 504–507, 2006.

[9] HINTON, Geoffrey E; ZEMEL, Richard S. Autoencoders, minimum description length and Helmholtz free energy. In: ADVANCES in neural information processing systems. [S.l.: s.n.], 1994. p. 3–10.

[10] KINGMA, Diederik P; WELLING, Max. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114, 2013.

[11] LECUN, Yann; BENGIO, Yoshua; HINTON, Geoffrey. Deep learning. nature, v. 521, n. 7553, p. 436, 2015.

[12] LE CUN, Yann et al. Handwritten digit recognition: Applications of neural network chips and automatic learning. IEEE Communications Magazine, IEEE, v. 27, n. 11, p. 41–46, 1989.

[13] LE CUN, Yann et al. Gradient-based learning applied to document recognition. Proceedings of the IEEE, IEEE, v. 86, n. 11, p. 2278–2324, 1998.

[14] MASCI, Jonathan et al. Stacked convolutional auto-encoders for hierarchical feature extraction. In: International Conference on Artificial Neural Networks. Springer, Berlin, Heidelberg, 2011. p. 52-59.

[15] NAGY, George. State of the art in pattern recognition. Proceedings of the IEEE, v. 56, n. 5, p. 836-863, 1968.

[16] Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. International Conference on Medical image computing and computer-assisted intervention. Springer, Cham, p. 234-241, 2015.

[17] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. Learning internal representations by error propagation. In Rumelhart, D. E. and McClelland, J. L., editors, Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 1: Foundations, MIT Press, Cambridge, MA. pp 318-362. 1986.

[18] SRIVASTAVA, Nitish et al. Dropout: a simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research, JMLR. org, v. 15, n. 1, p. 1929–1958, 2014.

[19] SZEGEDY, Christian et al. Inception-v4, inception-resnet and the impact of residual connections on learning. In: Thirty-First AAAI Conference on Artificial Intelligence. 2017.

[20] ZEILER, Matthew D.; FERGUS, Rob. Visualizing and understanding convolutional networks. In: European conference on computer vision. springer, Cham, p. 818-833, 2014.