

ACH2016 - Inteligência Artificial

Trabalho de Redes Neurais

Sarajane Marques Peres, André Paulino de Lima

Universidade de São Paulo

24 de março de 2017

Organização deste Documento

- 1 Objetivos e Datas Relevantes
- 2 Especificação da Entrega 1
- 3 Especificação da Entrega 2

Do Objetivo

Objetivo

Uma experimentação com Multilayer Perceptron (MLP) que consistirá na implementação de um sistema capaz de analisar caracteres escritos à mão e associá-los à uma classe pré-definida.

- Para ter uma ideia do assunto tratado no trabalho, assista o vídeo em <https://quickdraw.withgoogle.com/#>
- Não utilizaremos as mesmas técnicas empregadas pelos autores deste jogo (classificação de esboço desenhado à mão em modo *online*), mas a ideia é desenvolver uma rede neural que seja capaz de identificar a classe de um caractere escrito à mão (em modo *offline*).

Das Datas

Datas de Entrega

A data máxima para a Entrega 1 é **8 de maio**.

A data máxima para a Entrega 2 é **19 de junho**.

- Todos os artefatos que compõem uma entrega, descritos nas próximas seções, deverão ser postados no e-Tidia. Uma atividade específica para cada entrega será publicada no e-Tidia.
- Para as entregas, os alunos devem se organizar em grupos com no máximo 4 pessoas.¹
- Atentem para a postagem na última hora: múltiplos usuários tentando submeter ao mesmo tempo podem causar sobrecarga do servidor.
- Entregas após a data não serão aceitas. Integrantes de grupo que não submeter uma entrega até sua data limite receberão nota zero na entrega em questão.

¹ Evite trocar a composição do grupo no decorrer do desenvolvimento do trabalho.

Da Avaliação

Processo de Avaliação

A avaliação será realizada sobre cada um dos artefatos que compõem uma entrega: relatório, vídeo e código, nesta ordem de importância.

- Os pesos de cada artefato na nota final do trabalho são: relatório tem peso 70%, o vídeo tem peso 30% e o código junto com arquivos gerados na execução têm peso 10%. Se bem feito, pode servir para arredondamentos de nota.
- Material de terceiros em domínio público (mantidos em sítios acadêmicos ou sítios especializados no assunto) poderão ser usados sob a condição de estarem claramente referenciados no relatório. Falha em referenciar o uso de trabalho de terceiros caracteriza plágio. **Se for constatado plágio de qualquer natureza durante a avaliação do trabalho, os integrantes do grupo receberão nota zero.**
- Eventualmente, durante a avaliação dos trabalhos, os grupos podem ser chamados para esclarecer algum aspecto da entrega submetida.

Especificação da Tarefa 1

Entrega 1

A primeira entrega é composta do **código implementado pelo grupo** (e arquivos produzidos por sua execução em cenários distintos) e **um vídeo** explicando os aspectos mais relevantes do código, como descrito nos próximos slides.

- O grupo deverá projetar, implementar, testar e treinar uma rede MLP. O conjunto de dados a ser usado é especificado mais adiante. A arquitetura que deve ser adotada é *feedforward*, com 1 camada escondida, treinada com algoritmo *backpropagation*.
- A estratégia de seleção de parâmetros é livre. Recomendamos que se empregue uma amostra balanceada com no mínimo 30% das instâncias da partição de treinamento.²
- Para a estratégia de treinamento e avaliação dos modelos, recomendamos adotar *k-fold cross validation*, com k mínimo igual a 5 (veja Figura 1).

² Segundo testes realizados, escolhidos os parâmetros, o processamento do conjunto de dados e indução de modelo leva menos de 5min. ▶

Ilustração do *k-fold cross validation*

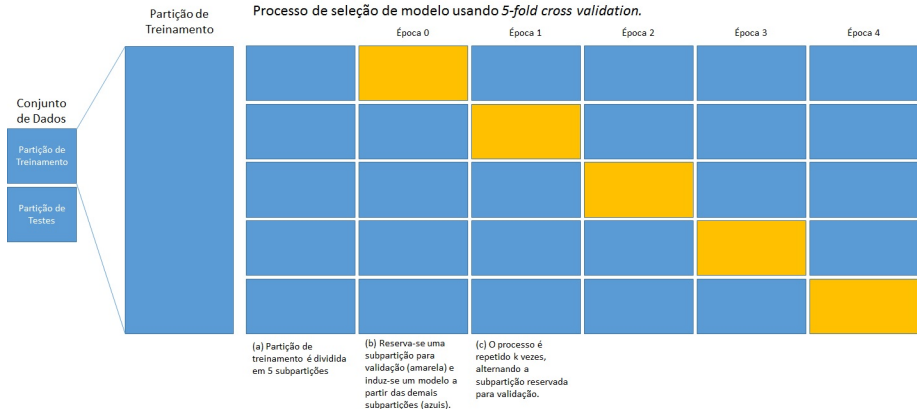


Figura 1: Processo de seleção de modelos usando *k-fold cross validation*, com k igual a 5.

Especificação da Tarefa 1 (Conjunto de Dados)

- R01** O conjunto de dados para esta tarefa é um subconjunto balanceado do '*NIST Special Database 19, 2nd Edition*', composto por **caracteres escritos à mão**, disponibilizados no formato PNG (*Portable Network Graphics*). O conjunto contém amostras dos caracteres **S**, **X** e **Z** (maiúsculos).
- O conjunto de dados está dividido em duas partições: (a) partição de treinamento, que conta com 1.000 instâncias para cada caractere (3.000 instâncias no total) e (b) partição de testes, que conta com 300 instâncias para cada caractere (900 instâncias no total).
 - Como comentamos acima, cada instância é disponibilizada em um arquivo PNG separado, representando uma imagem com 128 x 128 *pixels* RGB (cada pixel contém 3 valores de intensidade). Entretanto, como as imagens são codificadas em preto e branco, cada *pixel* assume, na verdade, apenas dois valores: branco (RGB=255,255,255) ou preto (RGB=0,0,0). Desta forma, cada instância pode ser reduzida a uma matriz de 128 x 128 *bits*, sem perda de informação.
 - Este conjunto de dados está disponível no e-Tidia (arquivo dataset1.zip).

Especificação da Tarefa 1

- O grupo deverá pré-processar o conjunto de dados e, para cada imagem, dois descritores diferentes devem ser extraídos: (a) descritor HOG (*Histogram of Oriented Gradients*), e (b) um outro descritor, a escolha do grupo. Algumas opções são listadas nas Referências.
- O grupo deverá induzir modelos específicos para cada um dos descritores acima, observando que a mesma arquitetura e parâmetros sejam empregados em ambos os modelos³.
- A extração de descritores HOG de uma imagem é realizada por um processo com múltiplas etapas, como ilustra a Figura 2: (1) Um filtro é aplicado à imagem, para ressaltar as mudanças de intensidade; (b) a imagem é dividida em células de igual tamanho; (c) para cada célula, calcula-se a orientação da troca de intensidade para cada *pixel*; (d) aplicando-se uma codificação por faixas de variação, um histograma de orientações é criado, sumarizando as orientações dos *pixels* na célula. O descritor HOG da imagem inteira é uma concatenação dos histogramas obtidos para cada célula.
- Este trabalho não requer que os grupos conheçam esse processo com detalhes: ao contrário, recomendamos que se empregue alguma implementação livre (*open source*) de extratores desse descritor (algumas são mencionadas logo adiante).

³ Incluindo o número de entradas da rede, que corresponde ao número de características dos descritores extraídos.

Ilustração do processo de extração de descritores HOG

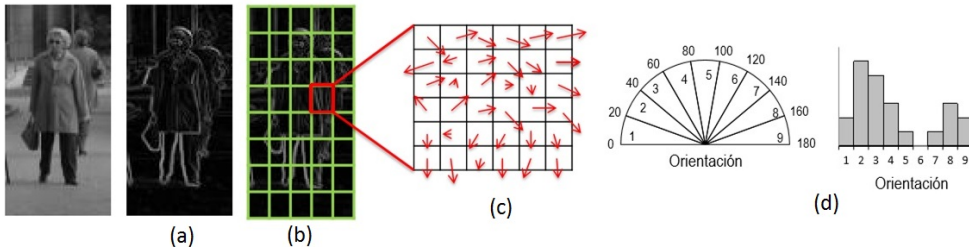


Figura 2: Processo de extração de descritores HOG. Extraído do material do *MOOC "Detección de Objetos"*, oferecido por Antonio M. López e Maria Vanrell, afiliados à *Universitat Autònoma de Barcelona*. Para simplificação da explicação, consideramos a blocagem unitária (blocos de 1x1 células).

Exploração do Conjunto de Dados (MATLAB)

Você pode usar o Matlab ou Python para ver como uma instância é representada numericamente e verificar que as três componentes da imagem (RGB) são idênticas.

MATLAB

```
A = imread('train_5a_00007.png');  
image(A);  
size(A)  
a1 = A(:, :, 1);  
a2 = A(:, :, 2);  
a3 = A(:, :, 3);  
isequal(a1, a2)  
isequal(a1, a3)
```

Exploração do Conjunto de Dados (Python)

Python com package scikit-image

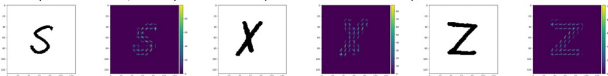
```
import numpy as np
import matplotlib.pyplot as plt
from skimage.io import imread, imshow
A = imread("train_5a_00007.png")
imshow(A); plt.show(A)
A.shape
a1 = A[:, :, 0]
a2 = A[:, :, 1]
a3 = A[:, :, 2]
np.array_equal(a1, a2)
np.array_equal(a1, a3)
```

Exploração do Conjunto de Dados

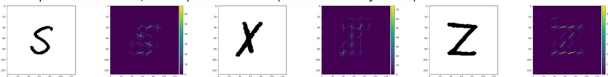
- Se você analisar imagens no conjunto de dados, logo vai perceber que há uma desproporcionalidade entre o fundo da imagem (em branco) e o desenho do caractere em si (em preto): há muito mais *pixels* representando o fundo do que representando o caractere. Isto sugere que talvez seja possível reduzir ainda mais a dimensionalidade da matriz que representa cada imagem, sem perda significativa de informação. Este processo é comumente referido como extração de descritores (ou características) da imagem.
- Existem várias abordagens para a extração de descritores (como LBP, Histograma de Blocos, HOG e Haar) e diversas implementações abertas destes extratores estão disponíveis para diferentes ambientes de programação. Exemplos: OpenCV (para Java, Python, C, e C++) ou scikit-image (para Python).
- A sensibilidade dos descritores é parametrizável. Por exemplo, como ilustra a Figura 3, extratores de descritores HOG costumam receber como parâmetro (a) o número de orientações; (b) o tamanho das células, em pixels; (c) o tamanho dos blocos, em células e (d) especificação de normalização, por exemplo.

Parâmetros do extrator de descritores HOG

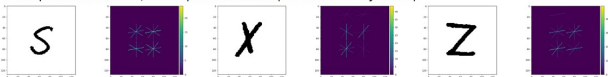
Pixels por célula = 8 x 8, células por bloco = 1 x 1 (vetores com 2.304 features)



Pixels por célula = 16 x 16, células por bloco = 1 x 1 (vetores com 576 features)



Pixels por célula = 32 x 32, células por bloco = 1 x 1 (vetores com 144 features)



Pixels por célula = 64 x 64, células por bloco = 1 x 1 (vetores com 36 features)

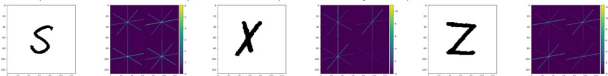


Figura 3: Ilustração do efeito do tamanho da célula na sensibilidade do descritor HOG. Usando 9 orientações.

Python com scikit-image

```
A = imread(sample)
a1 = A[:, :, 0]
(v, B) = hog(a1)
```

```
imshow(A)
plt.show()
```

```
imshow(B)
plt.show()
```

Especificação da Tarefa 1 (Artefatos - Vídeo)

Vídeo

O objetivo do vídeo é apresentar as características da codificação construída. **Todos os itens devem ser explicados usando o código-fonte do seu trabalho.**

Requisitos para o vídeo.

- R02 Duração de 10 a 15 min, formato MP4, resolução suficiente para o código estar legível.
- R03 Apresentar como se deu a extração de características das instâncias no conjunto de dados.
- R04 Apresentar quais são e como se configura os parâmetros (incluindo tipo de descritor).
- R05 Apresentar a estratégia de seleção de modelos (*5-fold cross validation*, por exemplo).
 - Apresentar as condições de aprendizado da rede: estratégia de cálculo do erro [R06], estratégia de cálculo da informação de erro [R07], estratégia de ajuste de pesos [R08], estratégia de decaimento de taxa de aprendizado [R09], verificação da condição de parada [R10], e outras características que você implementou que sejam importantes para valorizar seu trabalho.

Especificação da Tarefa 1 (Artefatos - Código)

- Para as entregas, serão aceitos códigos desenvolvidos nos seguintes ambientes de programação: R, Matlab, Python, Java, C e C++. Alinhar com o professor caso queira utilizar outro ambiente.
- Recomendamos que o grupo considere, ao selecionar o ambiente de programação, a disponibilidade de bibliotecas que ofereçam serviços básicos de manipulação de imagens em formato PNG e de extração de descritores, como HOG ou LBP.
- O código produzido deve estar bem comentado, de modo que o avaliador possa encontrar rapidamente os trechos referentes às principais funcionalidades (como leitura de parâmetros de configuração, implementações de funções de ativação e de erro, método de inicialização dos pesos, algoritmo de aprendizado e critérios de parada).

Especificação da Tarefa 1 (Artefatos - Código)

- R11 Ao executar, a implementação deve salvar em arquivo texto a configuração dos parâmetros que está sendo considerada (arquivo config.txt).
- R12 Ao executar, a implementação deve salvar em arquivo texto a progressão dos erros de treinamento, de validação e de testes ao longo das épocas (arquivo error.txt).
- R13 Ao executar, a implementação deve salvar os parâmetros do modelo treinado, de modo que o grupo ou o avaliador possa aplicar novas instâncias ao modelo treinado (arquivo model.dat). O formato do arquivo é de escolha do grupo, uma vez que serviços de serialização de objetos dependem do ambiente de programação adotado.
- R14 O grupo deve submeter os arquivos produzidos pelas execuções, organizados conforme ilustram as Figuras 4 e 5. A entrega mínima esperada deve conter esses arquivos para 4 cenários distintos: (a) modelo de melhor desempenho treinado com o descritor HOG; (b) modelo de pior desempenho treinado com o descritor HOG; (c) modelo de melhor desempenho treinado com o descritor escolhido pelo grupo e (d) modelo de pior desempenho treinado com o descritor escolhido pelo grupo.

Especificação da Tarefa 1 (Artefatos - Código)



Figura 4: Estrutura dos arquivos produzidos pela execução do código nos 4 cenários exigidos.

Especificação da Entrega 1

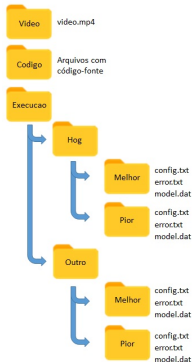


Figura 5: Organização dos artefatos para submissão pelo e-Tidia.

R15 A entrega é composta de um único arquivo, nomeado grupo_DD.zip, organizado como ilustra a Figura 5.

- O grupo pode submeter arquivos de execução referentes a outros cenários além dos mínimos, mas devem nomear as pastas com identificadores diferentes (e sem acentuação). Além disso, para essas execuções complementares, não submeter os arquivos model.dat, por questões de volume de armazenamento.
- As pastas com os arquivos de execução devem conter somente os arquivos listados.
- Atenção para o formato dos arquivos (cabeçalho e linhas de detalhe).

Especificação da Tarefa 2

- A segunda entrega deverá incluir como artefatos o código implementado pelo grupo (e arquivos produzidos por sua execução em distintos cenários) e o relatório final, analisando os resultados observados e suas justificativas, conforme entendimento do grupo.
- R16 Usando a mesma arquitetura adotada na Tarefa 1, o grupo deverá recalibrar a MLP e treiná-la usando um novo conjunto de dados. A recalibração pode modificar quaisquer parâmetros da rede, mas o grupo deve justificar a decisão no relatório (como mudar o número de entradas ou de neurônios da camada escondida).
- Para a estratégia de seleção de modelos, mantemos a recomendação de adoção de *k-fold cross validation*, com k mínimo igual a 5. Ainda, a separação das instâncias entre partições de treinamento e de testes deve ser respeitada.
 - O conjunto de dados precisará ser pré-processado para execução da extração dos descritores das instâncias. O grupo deverá produzir dois modelos: (a) modelo induzido a partir de descritores HOG e (b) modelo induzido a partir de descritores extraídos pelo filtro selecionado pelo grupo na Tarefa 1.

Especificação da Tarefa 2 (Conjunto de Dados)

- R17** O conjunto de dados para esta tarefa é um subconjunto balanceado do '*NIST Special Database 19, 2nd Edition*', composto de instâncias de **caracteres escritos à mão**, disponibilizados no formato PNG (*Portable Network Graphics*). O conjunto contém amostra dos 26 caracteres maiúsculos do alfabeto romano.
- O conjunto de dados está dividido em duas partições: (a) partição de treinamento, que conta com 1.000 instâncias para cada caractere (26.000 instâncias no total) e (b) partição de testes, que conta com 300 instâncias para cada caractere (7.800 instâncias no total).
 - A estrutura das instâncias é a mesma descrita na Tarefa 1.
 - Este conjunto de dados está disponível no e-Tidia (arquivo dataset2.zip).
 - No espírito da aplicação apresentada em <https://quickdraw.withgoogle.com/#>, o grupo pode criar e submeter suas próprias imagens para testar o desempenho da rede treinada. Para melhores resultados, as imagens devem ser capturadas em um canvas com 128 x 128 bits.

Especificação da Tarefa 2 (Artefatos - Relatório)

- R18 O relatório deverá ser elaborado seguindo o formato IEEE, disponível [neste link](#), opção '*Template and Instructions on How to Create Your Paper*'. As seções sugeridas não precisam ser seguidas: a ideia é usar a mesma diagramação, tamanho e tipo de fonte, estilo dos parágrafos, margens, referências bibliográficas, etc. O arquivo deve ser convertido no formato PDF antes da submissão da entrega.
- R19 O relatório deve apresentar aspectos da arquitetura selecionada e descrever seus parâmetros (como número de entradas, número de neurônios em cada camada, tipo de função de ativação de cada camada, função de custo (ou de erro) aplicada na saída da rede, método de inicialização dos pesos, passo de aprendizado e critérios de parada).
- R20 O relatório deve apresentar o método utilizado para seleção dos valores adotados para os parâmetros da rede; se houver inspiração em outros trabalhos publicados, cite-os adequadamente.
- R21 O relatório deve apresentar as curvas com a evolução dos erros de treinamento, de validação e de testes por época, para os casos mais interessantes de aprendizado (melhores e piores).

Especificação da Tarefa 2 (Artefatos - Relatório)

- R22 O relatório deve ainda apresentar a acurácia média por classe (caractere) e a matriz de confusão, obtidas nos testes dos dois modelos gerados (modelo treinado com descritores HOG e o modelo treinado com descritor selecionado pelo grupo).
- R23 O relatório deve apresentar uma análise comparativa dos resultados obtidos pelos modelos induzidos a partir de diferentes descritores, justificando a diferença em termos de influência dos parâmetros.
- R24 O relatório deve apresentar uma análise comparativa dos resultados obtidos pelos modelos com melhor e pior desempenho, justificando em termos de influência dos parâmetros.
 - O relatório não deve conter seções de explicações sobre a teoria referente aos algoritmos usados no trabalho. Nem tão pouco sobre as questões de implementação.

Especificação da Tarefa 2 (Artefatos - Código)

- Não é esperado que o grupo precise alterar o código entregue na Tarefa 1, pelo menos não de forma significativa.
- R25 Desta forma, seguir as mesmas recomendações sobre geração de arquivos de execução (arquivo config.txt, error.txt e model.dat) descritas para a Tarefa 1.
- R26 O grupo deve submeter os arquivos produzidos pelas execuções, organizados conforme ilustram a Figura 6. A entrega mínima esperada deve conter esses arquivos para 4 cenários distintos: (a) modelo de melhor desempenho usando descritores HOG; (b) modelo de pior desempenho usando descritores HOG; (c) modelo de melhor desempenho usando descritores escolhidos pelo grupo e (d) modelo de pior desempenho usando descritores escolhidos pelo grupo.

Especificação da Entrega 2

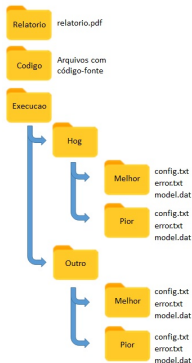


Figura 6: Organização dos artefatos para submissão pelo e-Tidia.

R27 A entrega é composta de um único arquivo, nomeado como grupo_DD.zip, organizado como ilustra a Figura 6.

- O grupo pode submeter arquivos de execução referentes a outros cenários além dos mínimos, mas devem nomear as pastas com identificadores diferentes (e sem acentuação). Além disso, para essas execuções complementares, não submeter os arquivos model.dat, por questões de volume de armazenamento.
- As pastas com os arquivos de execução devem conter somente os arquivos listados.
- Atenção para o formato dos arquivos (cabeçalho e linhas de detalhe).

Referências

O conjunto de dados NIST SD-19, *release* 2 (2016):

<https://www.nist.gov/srd/nist-special-database-19>

Bibliotecas para manipulação de imagens

- OpenCV: Open Source Computer Vision, disponível em <http://opencv.org/>, com interfaces para C++, C, Python, Java e MATLAB, sobre Windows, Linux, Mac OS e Android.
- scikit-image: disponível em <http://scikit-image.org/>, com interface para Python. Os responsáveis pedem gentilmente para que o uso da biblioteca seja citado. <http://dx.doi.org/10.7717/peerj.453>.
- MATLAB possui um pacote com funções úteis em processamento de imagens: *Computer Vision System Toolbox*.

Referências

Extração de Características

- Artigo sobre o descritor HOG: Dalal, Navneet, and Bill Triggs. "Histograms of oriented gradients for human detection." Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. Vol. 1. IEEE, 2005. Disponível neste [link](#). Também temos disponível um vídeo de um dos autores (Dalal) falando sobre o descritor HOG, neste [link](#).
- Outros descritores de imagem comumente utilizados em Visão por Computador: LBP (*Local Binary Patterns*), Filtros de Haar, SIFT (*Scale Invariant Feature Transform*), este último é patenteado nos EUA.
- Método empregado por Alpaydin e Kaynak para extração de características de instâncias do conjunto de dados NIST SD-19, *release 1* (1995): [link](#)
- Método empregado por LeCun, Cortes e Burges para extração de características de instâncias do conjunto de dados NIST SD-19, *release 1* (1995): [link](#).

Referências

Elementos Comuns da Arquitetura de Redes Neurais

- Funções de Ativação (também chamadas de *non-linearities* em textos em Inglês): degrau (step), sigmóide, tangente hiperbólica (Fausett, seções 1.4.3, 6.1.2), retificadora (artigo disponível neste [link](#)).
- Funções de Erro: MSE (*Mean Squared Error*, seção 5.1.4 do livro Deep Learning, disponível [online](#)); Softmax (seção 6.2.2.3 do mesmo livro).
- Métodos de inicialização: aleatória, pelo método de Nguyen-Widrow (Fausett, seção 6.1.2), Glorot e Bengio 2010 (veja o slide 26 deste [documento](#)). Para uma análise mais profunda sobre a controvérsia do assunto, veja este [artigo](#).
- Critério de parada antecipada usando cross-validation: Haykin, seção 4.13
- Maneira simples de reduzir o problema de sobreajuste (*overfitting*): procedimento de decaimento de pesos (*weight-decay procedure*), Haykin, seção 4.14.

Calibração de Parâmetros da Rede

A seleção de hiperparâmetros de um modelo pode ser uma tarefa desafiadora. Veja as dicas que Richard Socher, um pesquisador em redes neurais aplicadas ao processamento de linguagem natural, ofereceu em uma palestra na NAACL2013:

- Apresentação disponível neste [link](#) (de 1:18:25 a 1:34:20).
- Slides da apresentação podem ser encontrados neste [link](#), slides 177 a 190.
- Para este trabalho, ressaltamos a necessidade de se certificar que o modelo tem capacidade de sobreajuste ao conjunto de dados, e depois fazer os ajustes necessários para que isso não ocorra.

ACH2016 - Inteligência Artificial

Trabalho de Redes Neurais

Sarajane Marques Peres, André Paulino de Lima

Universidade de São Paulo

24 de março de 2017