SIN-5016 - Aprendizado de Máquina

Trabalho Final - Identificação e Autentificação Facial

Clodoaldo Ap Moraes Lima

Universidade de São Paulo

13 de julho de 2020

Organização deste Documento

Do Objetivo

Objetivo

Implementar 04 classificadores para a tarefa de reconhecimento facial. A tarefa de autenticação consiste em receber duas imagens e dizer se estas correspondem a mesma pessoa ou não. Já a tarefa de identificação consiste em receber uma imagem e dizer quem é esta pessoa.

Das Datas

Datas de Entrega

A data máxima para a Entrega é **30 de julho**.

- Todos os artefatos que compõem uma entrega, descritos nas próximas seções, deverão ser postados no e-Tidia
- Para as entregas, os alunos devem se organizar em grupos com no máximo 2 pessoas.[1]
- Atenção para a postagem na última hora: múltiplos usuários tentando submeter ao mesmo tempo, podem causar instabilidade no 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. Todas as entregas realizadas via email serão descartadas.

[1] Não será permitido grupos maiores.

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.

- A contribuição deste trabalho na nota final de cada aluno segue as regras informadas no esquema de avaliação, apresentado na primeira aula.
- 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 na entrega em questão.
- Eventualmente, durante a avaliação dos trabalhos, os grupos podem ser chamados para esclarecer algum aspecto da entrega submetida.

Especificação da Tarefa

Entrega 1

A entrega é composta do **código implementado pelo grupo** (e arquivos produzidos por sua execução em cenários distintos), **um relatório** explicando aspectos mais relevantes do código, como se descreve nos próximos slides, **um video** explicando a execução dos experimentos e detalhes gerais.

Especificação da Tarefa

- O grupo deverá empregar quatro classificadores (Rede MLP, SVM, Ensemble de Modelos Heterogêneos e CNN) usando o conjunto de dados especificado mais adiante. A arquitetura que deve ser adotada, no caso da MLP *feedforward*, consiste de uma rede com 1 camada escondida, treinada com algoritmo de aprendizado *backpropagation*, já no caso do SVM, deve ser o SVM tradicional (C-SVC).
- 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]
- No caso da MLP e CNN, a interrupção do processo de treinamento não precisa aguardar a conclusão de k épocas – o critério de parada antecipada adotado deve ter precedência sobre o processo de seleção de modelos.

[2] Segundo testes realizados, escolhidos os parâmetros, o processamento do conjunto de dados e indução de modelo deve se rápido.

Especificação da Tarefa

- Para a estratégia de treinamento e avaliação dos modelos, recomendamos adotar *k-fold cross validation*, com k igual a 5 (veja Figura 1).
- O grupo deverá pré-processar o conjunto de dados usando o algoritmo Viola & Jones 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. Recomendo que seja utilizado o LPB (Local Pattern Binary) ou Transformada Wavelet. Somente a implementação dos descritores pode ser utilizada de terceiros.
- O grupo deverá induzir modelos específicos para cada um dos descritores acima, observando que a mesma arquitetura e parâmetros sejam empregados em ambos os modelos[3].
- O conjunto de dados precisará ser pré-processado para extrair características das instâncias do conjunto de dados. O grupo deverá produzir dois modelos, usando a mesma arquitetura e parâmetros[4]: (a) modelo induzido a partir de descritores HOG (*Histogram of Oriented Gradients*), e (b) modelo induzido a partir de descritores extraídos por um outro filtro, a escolha do grupo.

[3] Incluindo o número de entradas, que corresponde ao número de características dos descritores extraídos.

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

Especificação da Tarefa

- 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).

Ilustração do *k-fold cross validation*

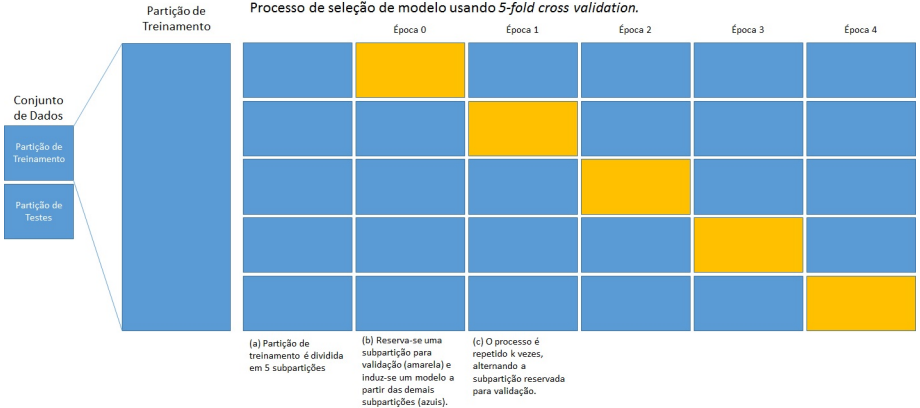

Figura 1: Processo de seleção de modelos usando *k-fold cross validation*, com k igual a 5.

Ilustração do processo de extração de descritores HOG

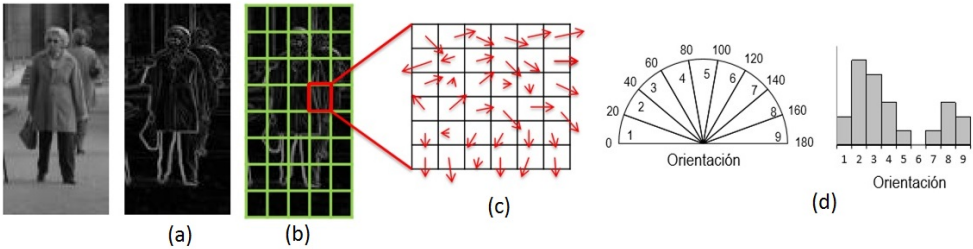

Figura 2: Processo de extração de descritores HOG. Extraído do material do *MOOC "Detección de Objetos"*, oferecido por Antonio M. López e Maria Vanrell, afiliados à *Universitat Autònoma de Barcelona*. Para simplificação da explicação, consideramos blocos de 1x1 células).

Especificação da Tarefa (Base de Dados)

- O aluno deve escolher uma das seguintes bases de dados: i) Base de dados LFW (http://vis-www.cs.umass.edu/lfw/index.html), Base de dados LFWA (https://talhassner.github.io/home/projects/lfwa/ ou Base de dados CelebA (http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html))
- A base de dados LFW contém um conjunto de imagens de face projetado para estudar o problema do reconhecimento facial em ambiente não controlado. Esta base de dados contém 13233 imagens de 5749 pessoas, 1680 pessoas com duas ou mais imagens. O objetivo é desenvolver um modelo para tarefa de classificação binária (autenticação ou verificação biométrica).
- Esta base contém as mesmas imagens disponíveis no conjunto de dados Original Labeled Faces in the Wild , no entanto, nesta as imagens foram alinhadas usando um software comercial de alinhamento de faces. O objetivo é desenvolver um modelo para tarefa de classificação binária (autenticação ou verificação biométrica).

Especificação da Tarefa (Base de Dados)

- A base de dados CelebFaces (CelebA) contém 202,599 imagens de celebridades, cada uma com 40 anotações de atributos. As imagens neste conjunto de dados cobrem grandes variações de pose e fundo. A base de dados pode ser empregada como conjunto de treinamento e teste para as seguintes tarefas de visão computacional: reconhecimento de atributo facial, detecção facial, localização de ponto de referência e edição e síntese de face. Nesta atividade, o objetivo é desenvolver um modelo para tarefa de classificação com múltiplas classes (com pelo menos 10% da base) (identificação biométrica).

Exploração do Conjunto de Dados

- 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 diversos 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 **??** 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.

Especificação da Tarefa (Relatório)

Vídeo

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

Requisitos para o relatório.

R01 Duração de 10 a 15 min, formato MP4, resolução suficiente para o código estar legível.

R02 Cada membro deve gravar um exemplo explicando a codificação realizada por 02 modelos

R03 Cada membro deve demonstrar conhecimento de todos os códigos desenvolvidos

Especificação da Tarefa (Relatório)

Vídeo

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

Requisitos para o relatório.

R01 Apresentar quais são e como se configura os parâmetros (incluindo tipo de descritor).

R02 Apresentar estruturas de dados que organizam os pesos que compõem as camadas da rede.

R03 Apresentar como se deu a extração de características das instâncias no conjunto de dados.

R04 Apresentar a estratégia de seleção de modelos (*5-fold cross validation*, por exemplo).

R05 Apresentar como atuam os algoritmos de inicialização de pesos e a implementação do algoritmo de treinamento da rede.

R06 Apresentar os resultados obtidos em forma de tabelas e gráficos.

R07 Analisar os resultados obtidos.

Especificação da Tarefa - (Artefatos - Relatório)

R08 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.

R09 O relatório deve apresentar aspectos dos modelo. No caso de uma rede neural, a 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).

R10 O relatório deve apresentar o método utilizado para seleção dos valores adotados para os parâmetros, se houver inspiração em outros trabalhos publicados, cite-os adequadamente.

R11 O relatório deve apresentar as curvas com a evolução dos erros de treinamento, de validação e de testes por época, conforme apresentado em sala.

Especificação da Tarefa (Artefatos - Relatório)

R12 O relatório deve ainda apresentar a acurácia média por classe (caractere), obtida nos testes para os dois descritores (modelo treinado com descritores HOG e o modelo treinado com descritor selecionado pelo grupo).

R13 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.

R14 O relatório deve apresentar uma análise comparativa dos resultados obtidos pelos modelos com melhor e pior desempenho, justificando em termos de influência dos parâmetros.

- Não é necessário explicar a teoria referente aos algoritmos usados no trabalho.

Especificação da Tarefa (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 (Artefatos - Código)

R13 Ao executar, a implementação deve salvar em arquivo texto a configuração dos parâmetros que está sendo considerada (arquivo config.txt).

R14 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).

R15 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.

R16 O grupo deve submeter os arquivos produzidos pelas execuções, organizados conforme ilustram as Figuras 3 e 4. 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 (Artefatos - Código)

Figura 3: Estrutura dos arquivos produzidos pela execução do código nos 4 cenários exigidos.

Especificação da Entrega

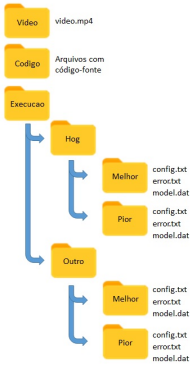

Figura 4: Organização dos artefatos para submissão pelo e-Tidia.

R17 A entrega é composta de um único arquivo, nomeado grupo_DD.zip, organizado como ilustra a Figura 4.

- 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 (Artefatos - Código)

- Não é esperado que o grupo precise alterar o código entregue na Tarefa 1, pelo menos não de forma significativa.

R18 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.

R19 O grupo deve submeter os arquivos produzidos pelas execuções, organizados conforme ilustram a Figura 5. 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

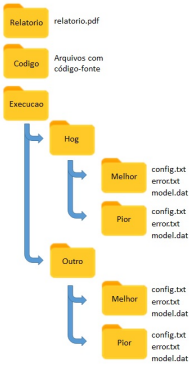

Figura 5: Organização dos artefatos para submissão pelo e-Tidia.

R20 A entrega é composta de um único arquivo, nomeado como 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).

Referências

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 sobdre 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 (Faussett, 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 as chances do modelo "*overfit*": 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 "*overfit*" o dataset, e depois fazer os ajustes necessários para que isso não ocorra.

SIN-5016 - Aprendizado de Máquina

Trabalho Final - Identificação e Autentificação Facial

Clodoaldo Ap Moraes Lima

Universidade de São Paulo

13 de julho de 2020