

# SIN-5016 - Aprendizado de Máquina

## Trabalho Final - Verificação Facial

Clodoaldo Ap Moraes Lima

Universidade de São Paulo

28 de Dezembro de 2021

# Organização deste Documento

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

## Do Objetivo

### Objetivo

Implementar 01 classificador baseado em Redes Neurais Convolucionais para a tarefa de verificação facial. A tarefa de verificação consiste em receber duas imagens e dizer se estas correspondem a mesma pessoa ou não.

# Das Datas

## Datas de Entrega

A data máxima para a Entrega é **20 de Janeiro**.

- Todos os artefatos que compõem uma entrega, descritos nas próximas seções, deverão ser postados no e-disciplina
- Para as entregas, os alunos devem se organizar em grupos com no máximo 2 pessoas.<sup>1</sup>
- 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.

<sup>1</sup> 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 vídeo** explicando a execução dos experimentos e detalhes gerais.

## Especificação da Tarefa

- O grupo deverá empregar 01 classificador baseado em Redes Neurais Convolucionais(CNN) usando o conjunto de dados especificado no EP 1. As arquiteturas que devem ser adotada são LexNet, LeNet todas com duas saídas (saídas [1 0] para imagens da mesma pessoa e [0 1] para imagens de pessoas diferentes). As rede devem receber duas imagens sem qualquer préprocessamento.
- Recomendo que se empregue uma amostra balanceada com no mínimo 30% das instâncias da partição de treinamento.<sup>2</sup>
- Os resultados obtidos devem ser comparados com os obtidos no trabalho parcial, em termos de taxa de classificação correta.

---

<sup>2</sup>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 10 (veja Figura 1).
- O conjunto de dados não precisará ser pré-processado para extrair características das instâncias do conjunto de dados.

# Ilustração do *k-fold cross validation*

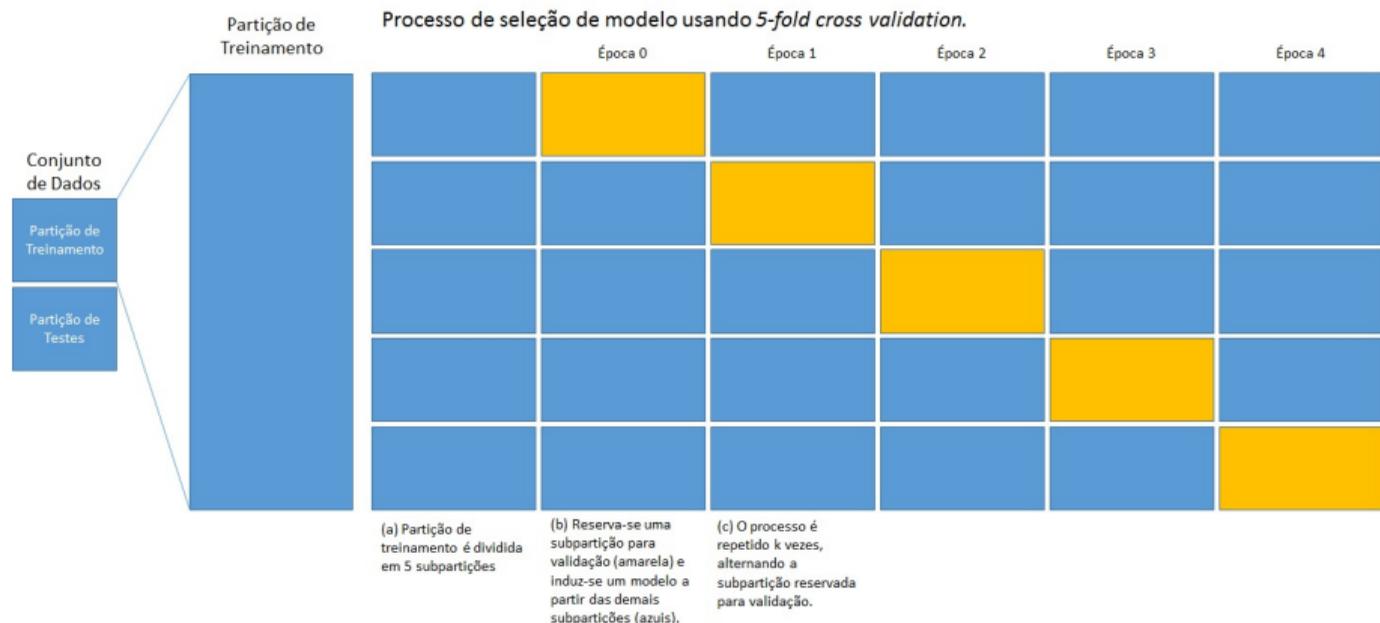


Figura 1: Processo de seleção de modelos usando *k-fold cross validation*, com  $k$  igual a 5.

## Especificação da Tarefa (Base de Dados)

- A base de dados a ser utilizada é a LFW (<http://vis-www.cs.umass.edu/lfw/index.html>).
- 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 25% da base) (identificação biométrica).

## 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 para pelo menos 01 experimento
- 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.
- R02 Apresentar estruturas de dados que organizam os pesos que compõem as camadas da rede.
- R03 Apresentar como atuam os algoritmos de inicialização de pesos e a implementação do algoritmo de treinamento da rede.
- R04 Apresentar os resultados obtidos em forma de tabelas e gráficos.
- R05 Analisar os resultados obtidos.

## Especificação da Tarefa - (Artefatos - Relatório)

- R06 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, tais como, como número de entradas, número de filtros 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)
- R13 O relatório deve apresentar uma análise comparativa dos resultados obtidos neste EP2 com os resultados obtidos no EP1
- R14 O relatório deve apresentar uma análise comparativa dos resultados obtidos pela CNN com melhor e pior desempenho, justificando em termos de influência dos parâmetros.
  - Não é necessário explicar a teoria referente aos algoritmos usados no trabalho.

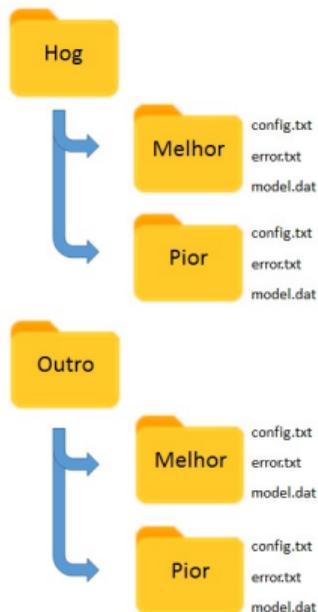
## Especificação da Tarefa (Artefatos - Código)

- 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
- 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 2 e 3. A entrega mínima esperada deve conter esses arquivos para 2 cenários distintos: (a) modelo de melhor desempenho; (b) modelo de pior desempenho.

# Especificação da Tarefa (Artefatos - Código)



run\_config.txt

O arquivo deve ter um cabeçalho seguido de linhas de detalhe:

Cabeçalho:

Execucao em 01/04/2017 14:00

Linha de detalhe: composta pelo nome do parâmetro e seu valor, separados por dois pontos. Números decimais representados com ponto (ao invés de vírgula).

Exemplo:

```
MLP_SPECIFICATON: ('layer 0', 14, 'sigmoid', 'mse')
MLP_SPECIFICATON: ('layer 1', 3, 'sigmoid', 'mse')

MLP_OPERATION_ETA_METHOD : FIX
MLP_OPERATION_ETA_PARAMS : 0.01
MLP_OPERATION_INITIALIZATION : Glorot_Bengio_2010
MLP_OPERATION_MAX_EPOCHS : 200
MLP_OPERATION_MIN_EPOCHS : 10
MLP_OPERATION_STOP_WINDOW : 6
```



run\_erroctxt

O arquivo deve ter um cabeçalho seguido de linhas de detalhe:

Cabeçalho:

Execucao em 01/04/2017 14:00

Linha de detalhe: colunas com dados de época, erro médio de treinamento (k-1 subpartições de treinamento), erro médio de validação (subpartição reservada para validação). Campos separados por ponto e vírgula. Números decimais representados com ponto (ao invés de vírgula).

Exemplo:

```
0;0.537532256;0.510597465
1;0.437917031;0.734788994
2;0.431760194;0.487446779
3;0.443171165;0.498086418
4;0.456150695;0.512672343
5;0.432404126;0.708678751
```

Figura 2: Estrutura dos arquivos produzidos pela execução do código nos 2 cenários exigidos.

# Especificação da Entrega

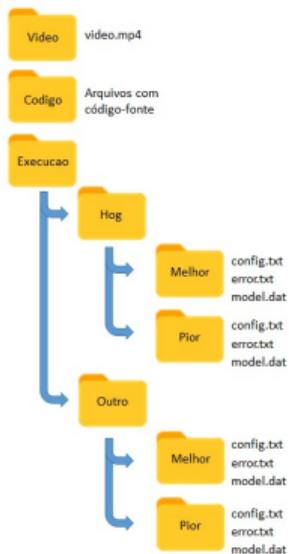


Figura 3: Organização dos artefatos para submissão pelo e-disciplina

R17 A entrega é composta de um único arquivo, nomeado grupo\_DD.zip, organizado como ilustra a Figura 3.

- 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)

- 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 4. A entrega mínima esperada deve conter esses arquivos para 2 cenários distintos: (a) modelo de melhor desempenho; (b) modelo de pior desempenho.

# Especificação da Entrega

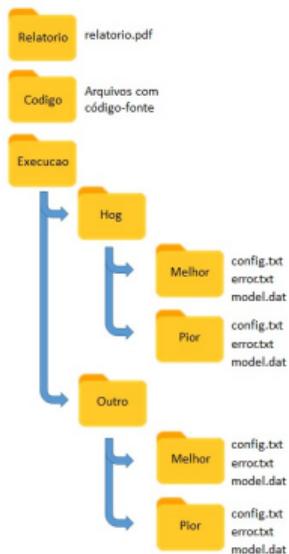


Figura 4: Organização dos artefatos para submissão pelo e-Tidia.

**R20** A entrega é composta de um único arquivo, nomeado como grupo\_DD.zip, organizado como ilustra a Figura 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).

# 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

## 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 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 - Verificação Facial

Clodoaldo Ap Moraes Lima

Universidade de São Paulo

28 de Dezembro de 2021