PSI5886 - Princípios de Neurocomputação
Prof Dr. Emílio Del Moral Hernandez
Escola Politécnica da USP

Transfer Learning

Felipe Marino Moreno

José Amendola Netto Andrade

Helio Takahiro Sinohara

Pedro Henrique Hauy Netto de Araujo

Grupo

José Amendola Netto Andrade
- Redes Neurais Convolucionais

Felipe Marino Moreno
- Arquiteturas de CNN

Pedro Henrique Hauy Netto de Araujo
- Transfer Learning

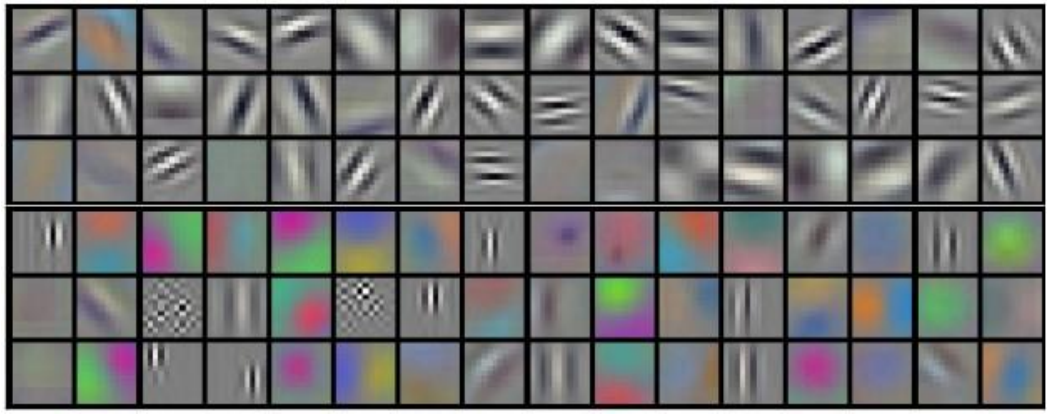
Helio Takahiro Sinohara
- Ensaios

Transfer Learning

- Objetivo do Trabalho:
  - **Analisar e realizar ensaios sobre Transfer Learning no treinamento de Redes Neurais Convolucionais para Classificação de Imagens**

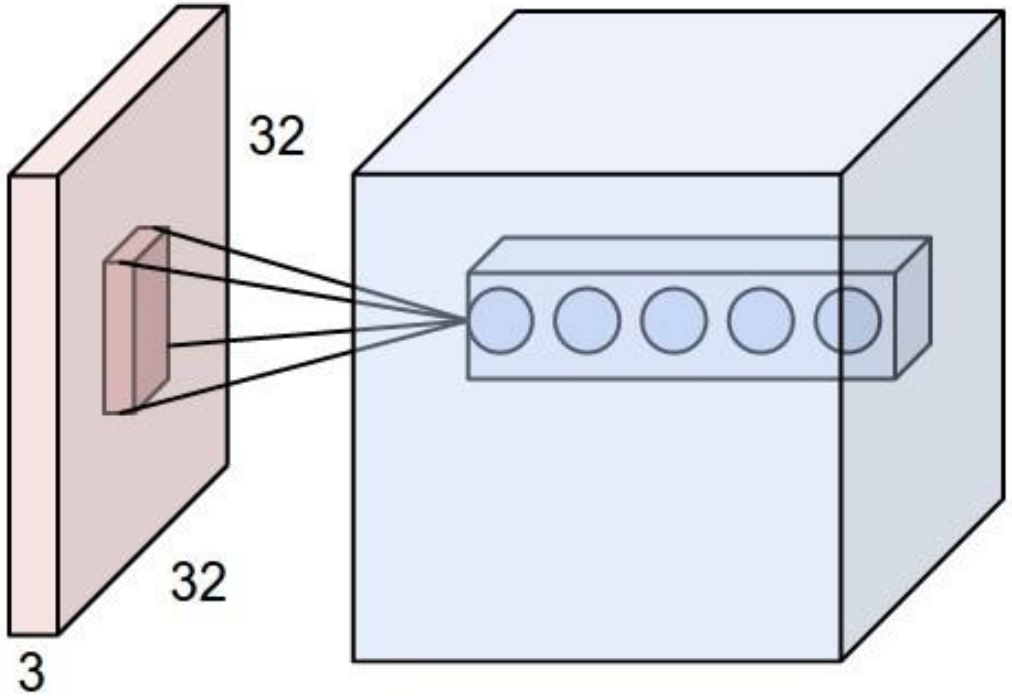
CNN’s: Revisão de conceitos básicos

- Neurônios e dados estruturados em forma 3D: comprimento, altura e profundidade
- Neurônios localmente conectados
- Compartilhamento de parâmetros dos neurônios, formando filtros (ou “kernels”)
- Hiperparâmetros:
  - **Número de filtros K**
  - **Tamanho do filtro F**
  - **Stride S**
  - **Zero-Padding P**



Fonte:http://cs231n.github.io/convolutional-networks/#norm

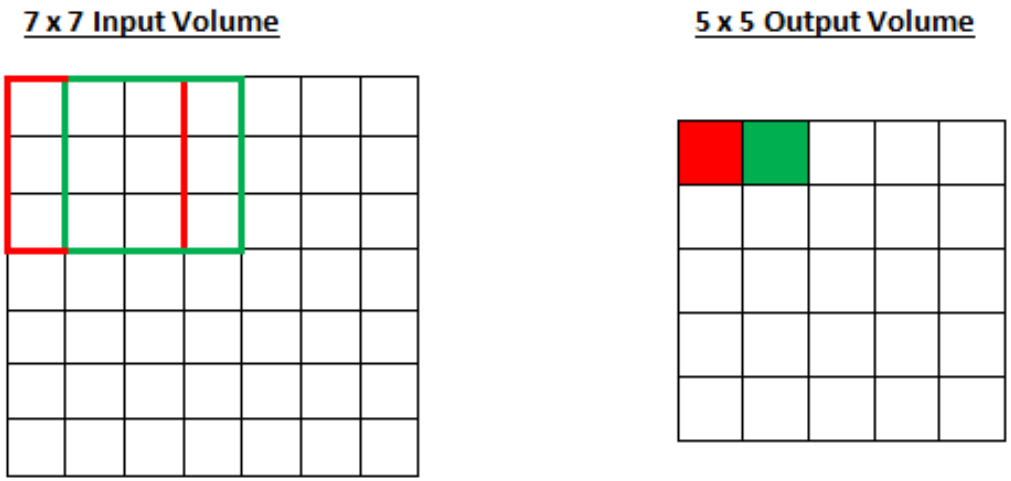
CNN’s: Revisão de conceitos básicos



Fonte:http://cs231n.github.io/convolutional-networks/#norm

Profundidade (depth): Quantidade de filtros convolucionais, cada qual visando algo diferente na imagem de entrada.

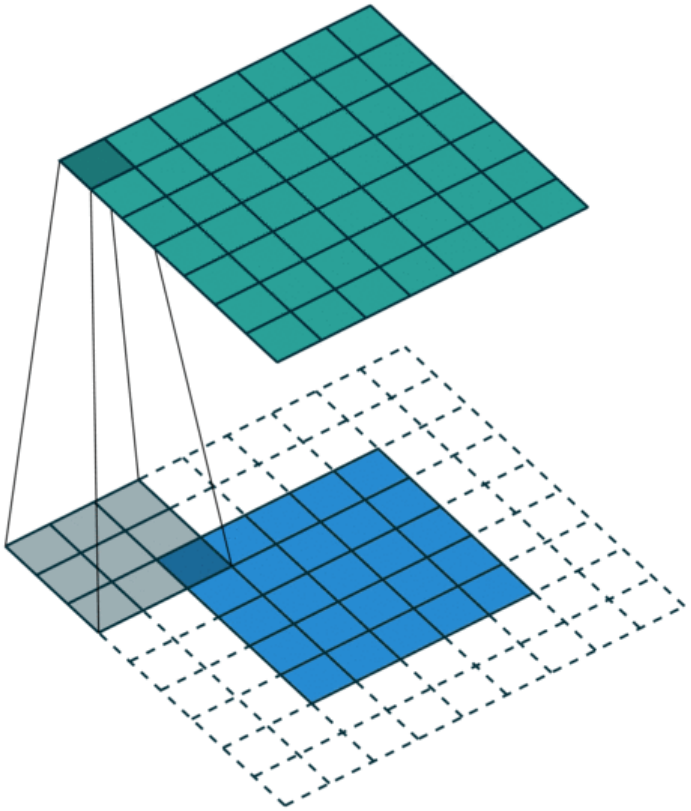
CNN's: Revisão de conceitos básicos



Fonte:http://cs231n.github.io/convolutional-networks/#norm

Stride: Número de pixels deslocados na varredura do filtro. Valores 1 e 2 comumente usados, 3 mais raro.

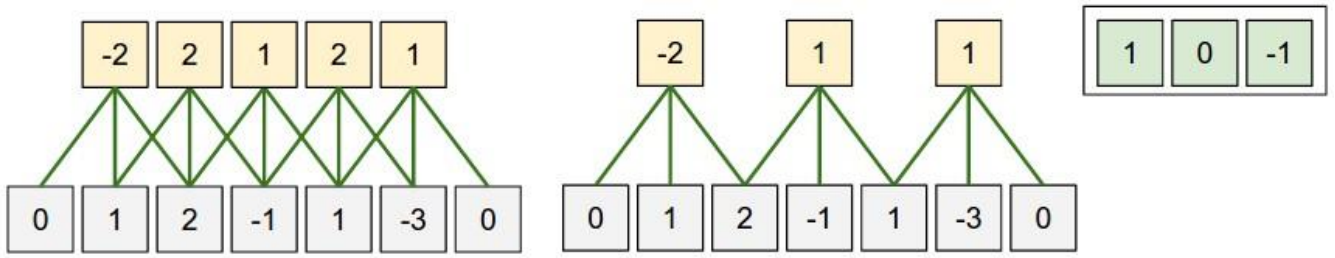
CNN's: Revisão de conceitos básicos



Zero-Padding: Tamanho da borda a ser preenchida com zeros

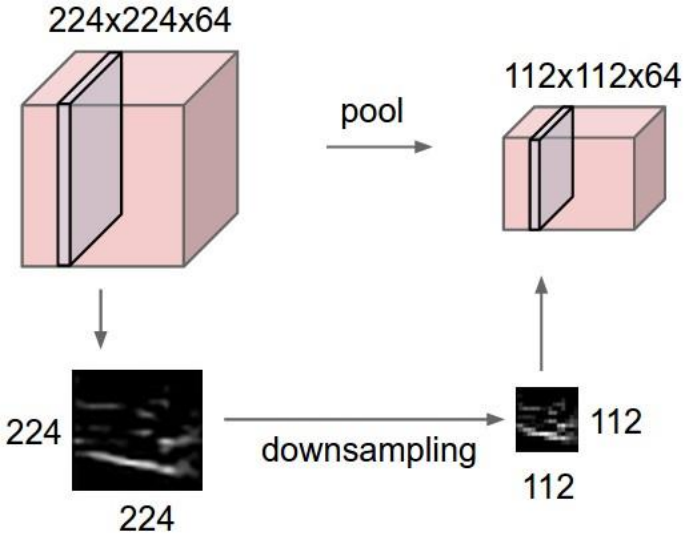
- Controla o tamanho do volume de dados ao longo das camadas, evitando que informações das bordas se percam.
- Wsaida = (W-F+2P)/(S+1)

Fonte:http://cs231n.github.io/convolutional-networks/#norm



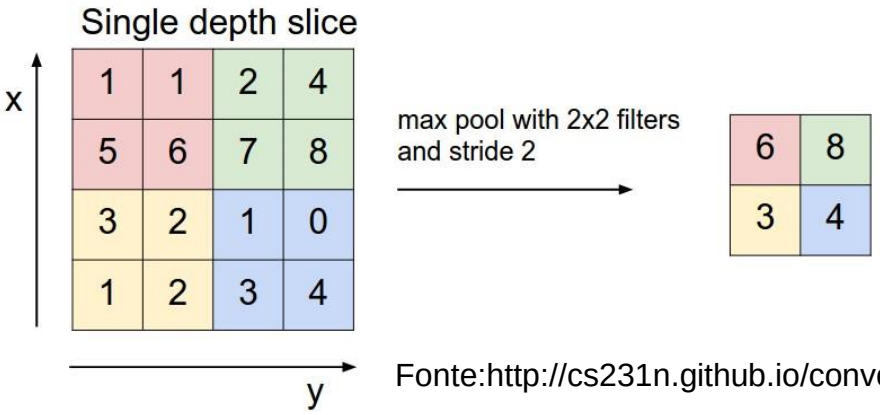
Fonte:http://cs231n.github.io/convolutional-networks/#norm

CNN's: Revisão de conceitos básicos



- **Pooling**: Amostragem feita em cada fatia do volume de dados resultante da camada convolucional.
- Altura e largura reduzidas, profundidade mantida.
- Também possui tamanho de filtro e stride.
- Max-Pooling: Variante cuja operação de filtro busca o maior valor dentro da região de filtragem.
- Há publicações que julgam as camadas de pooling desnecessárias [1]

Fonte:http://cs231n.github.io/convolutional-networks/#norm



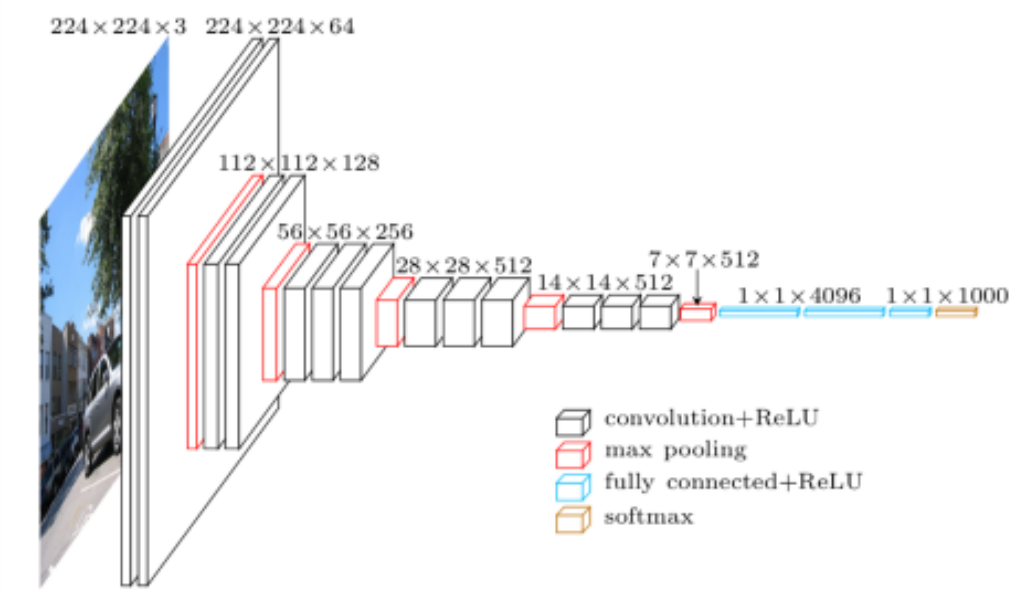
Fonte:http://cs231n.github.io/convolutional-networks/#norm

Redes Utilizadas

- Transfer Learning: VGG16
  - Desenvolvida para a competição da IMAGENET em 2014
  - Rede profunda com 16 camadas
  - Camadas de convolução disponíveis já pré-treinadas com o banco de dados da IMAGENET
  - Entrada 224x224x3 pixels
- Classificação de placas de trânsito: LeNet
  - Desenvolvida em 1998 para classificação de caracteres escritos manualmente
  - Rede convolucional relativamente simples, treinamento rápido.
  - Entrada de 32x32x1 pixels

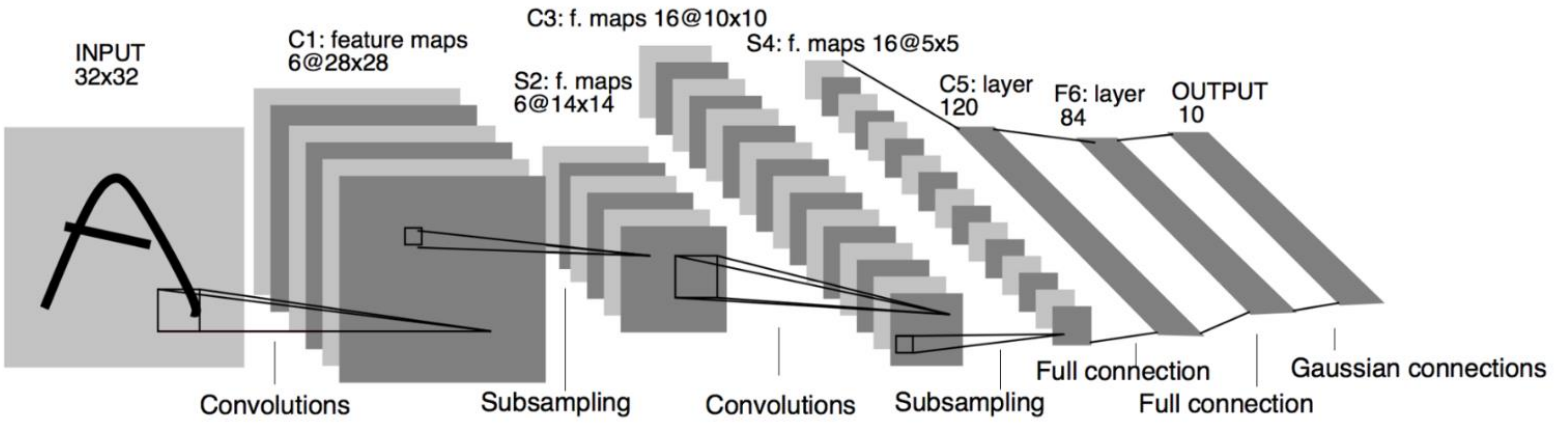
VGG16

- Convolução:
  - 13 camadas
  - Kernel ou filtro: 3x3
  - Padding: 1 pixel na borda
  - Stride: 1 pixel
  - Max pooling: 2x2 pixels
- Fully Connected
  - 3 camadas
  - Originalmente para classificação com 1000 labels
  - Camadas que serão substituídas no nosso ensaio



Fonte: https://arxiv.org/pdf/1409.1556.pdf

LeNet-5



Fonte: retirada de [8].

- 3 camadas de convolução (C1,C3,C5)
  - Kernel ou filtros: 5x5 pixels
  - Stride: 1 pixel
  - Average pooling: 2x2 pixels (S2,S4)
  - C5 fully-connected
  - C3 não utiliza todos os features da camada S2

Transfer Learning

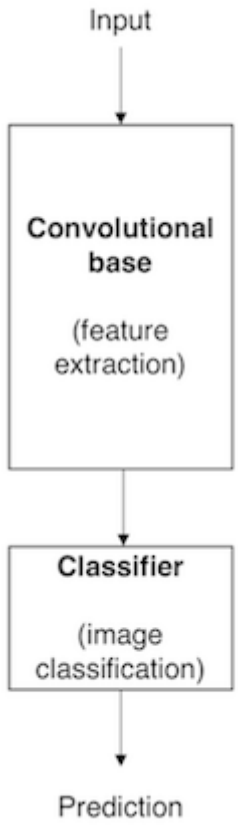
- O processo de utilizar o um modelo de Machine Learning em um contexto diferente (mas relacionado) do qual ele foi concebido [2]
- Pode ser reutilizada apenas a arquitetura ou também os pesos do modelo pré-treinado
- Útil quando não se dispõe dos mesmos recursos do problema original (tamanho do dataset, poder computacional, tempo de prototipagem).
- Ainda assim, possui benefícios mais gerais.

[2] Transfer Learning – Towards Data Science. [online] Disponível em https://towardsdatascience.com/transfer-learning-946518f95666 [Acesso em 26 Nov. 2018].

Aplicações de Transfer Learning

- Processamento de Linguagem Natural [3]
  - word2vec (Google)
  - GloVe (Stanford)

- Processamento de Imagens [3]
  - Modelos se popularizaram após a competição ImageNet
  - Redes Neurais Convolucionais (ilustrada ao lado)
    - LeNet
    - VGG16



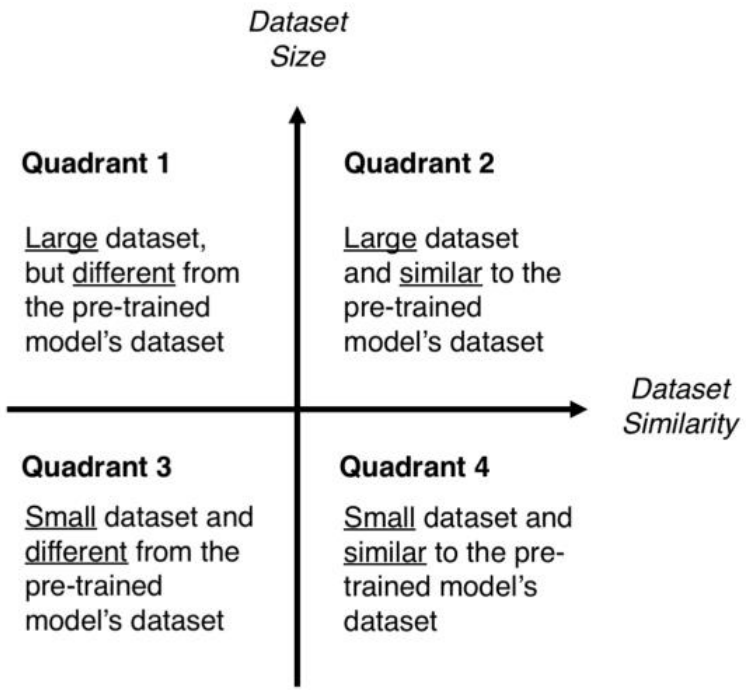
Fonte: retirada de [4].

[3] A Gentle Introduction to Transfer Learning for Deep Learning. [online] Disponível em: https://machinelearningmastery.com/transfer-learning-for-deep-learning [Acesso em 26 Nov. 2018].

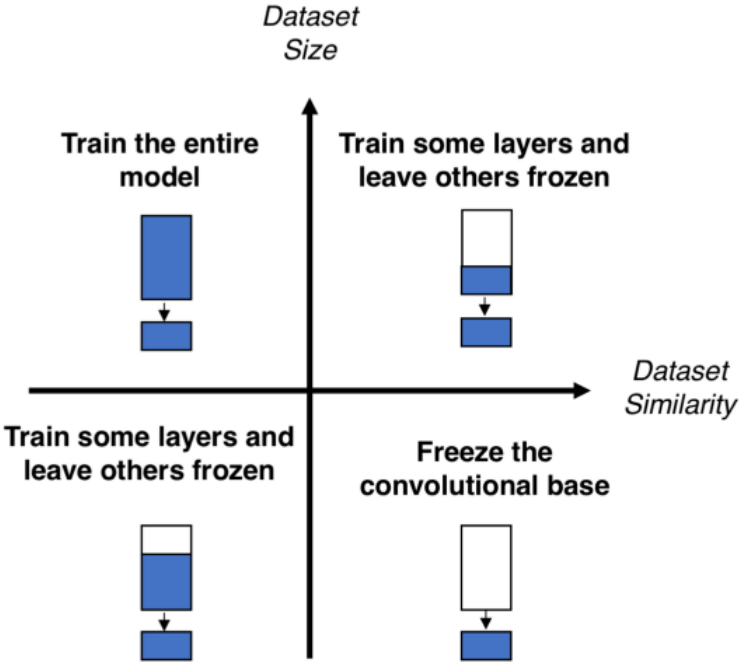
[4] Transfer learning from pre-trained models – Towards Data Science. [online] Disponível em: https://towardsdatascience.com/transfer-learning-from-pre-trained-models-f2393f124751 [Acesso em 26 Nov. 2018].

Estratégias de Implementação

- Dependendo do tamanho do Dataset de destino e sua similaridade com o de origem, diferentes estratégias podem ser adotadas [4].



Fonte: retirada de [4].



Fonte: retirada de [4].

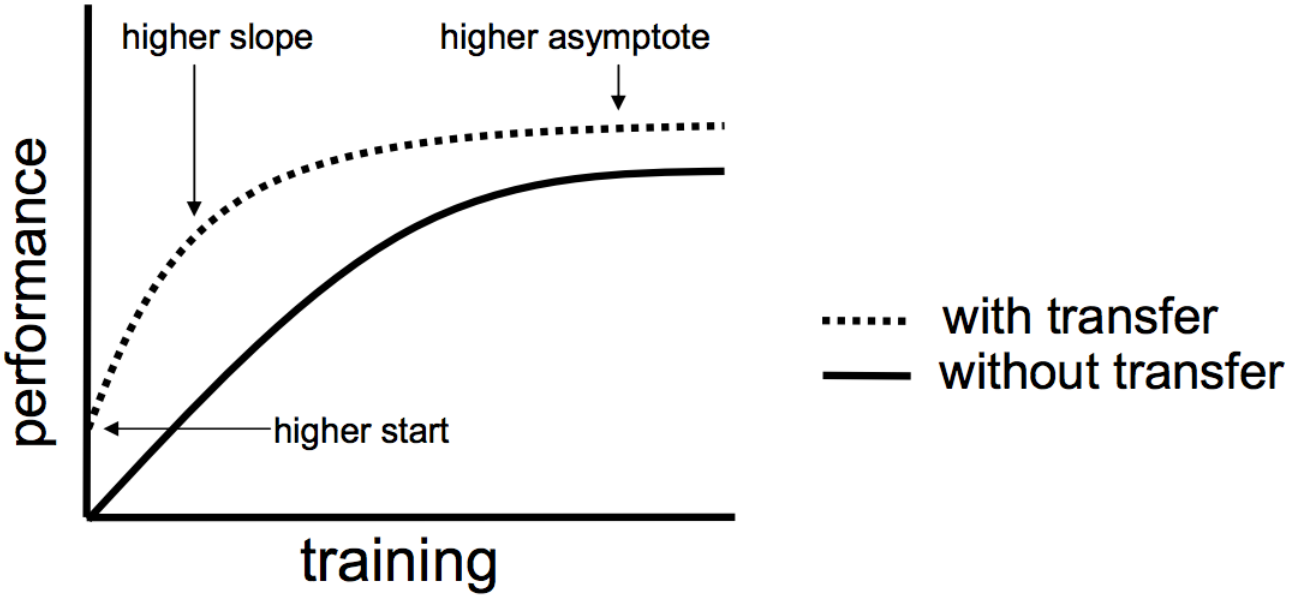
- Existem propostas de métricas de similaridade entre tarefas [5], mas para o caso de regressões (não classificadores).

[4] Transfer learning from pre-trained models – Towards Data Science. [online] Disponível em: https://towardsdatascience.com/transfer-learning-from-pre-trained-models-f2393f124751 [Acesso em 26 Nov. 2018].

[5] Cao, B., Pan, S. J., Zhang, Y., Yeung, D. Y., & Yang, Q. (2010, July). **Adaptive Transfer Learning**. In AAAI (Vol. 2, No. 5, p. 7).

Ganhos com Transfer Learning

- Mesmo em situações teoricamente já mais “favoráveis”, há benefícios com o uso de Transfer Learning [6].
- A rede tende a generalizar melhor e aprender mais rápido, partindo de uma melhor performance inicial [7].



Fonte: retirada de [7].

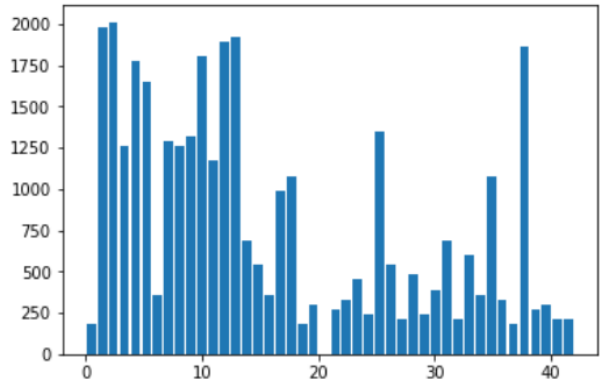
[6] JasonYosinski, JeffClune, YoshuaBengio, and HodLipson, “How Transferable are Features in Deep Neural Networks?” In Advances in Neural Information Processing Systems 27 (NIPS ’14), NIPS Foundation, 2014.
[7] Soria Olivas, E. (2010). “Handbook of research on machine learning applications and trends”. Hershey, Pa.: IGI Global (701 E. Chocolate Avenue, Hershey, Pennsylvania, 17033, USA).

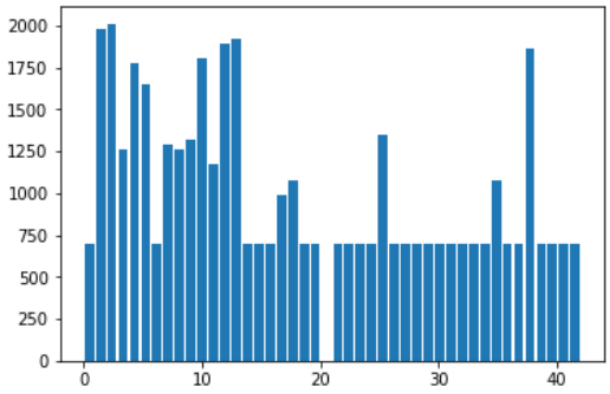
Ensaios- LeNet

```
Number of training examples = 34799
Number of testing examples = 12630
Image data shape = (32, 32, 3)
Number of classes = 43
validation samples = 4410
```

German Traffic Sign Dataset (http://benchmark.ini.rub.de/?section=gtsrb&subsection=dataset)

Antes do Pré-processamento

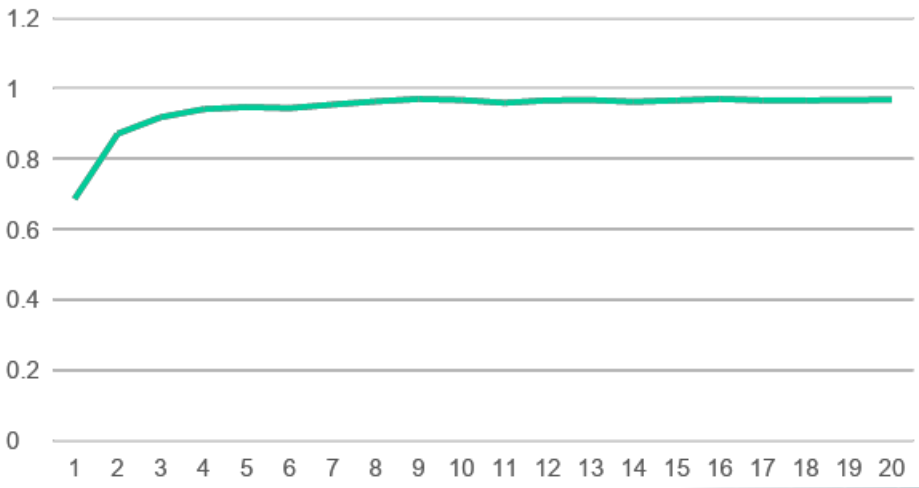




Após o Pré-processamento: Blur, rotações e normalização.

Ensaios - LeNet

Dropout = 0.7 (mantém 70% das conexões)
EPOCHS = 20
BATCH_SIZE = 128
Learning rate = 0.001
Otimizador ADAM
Data Augmentation:
Rotação, Blur, Equalização

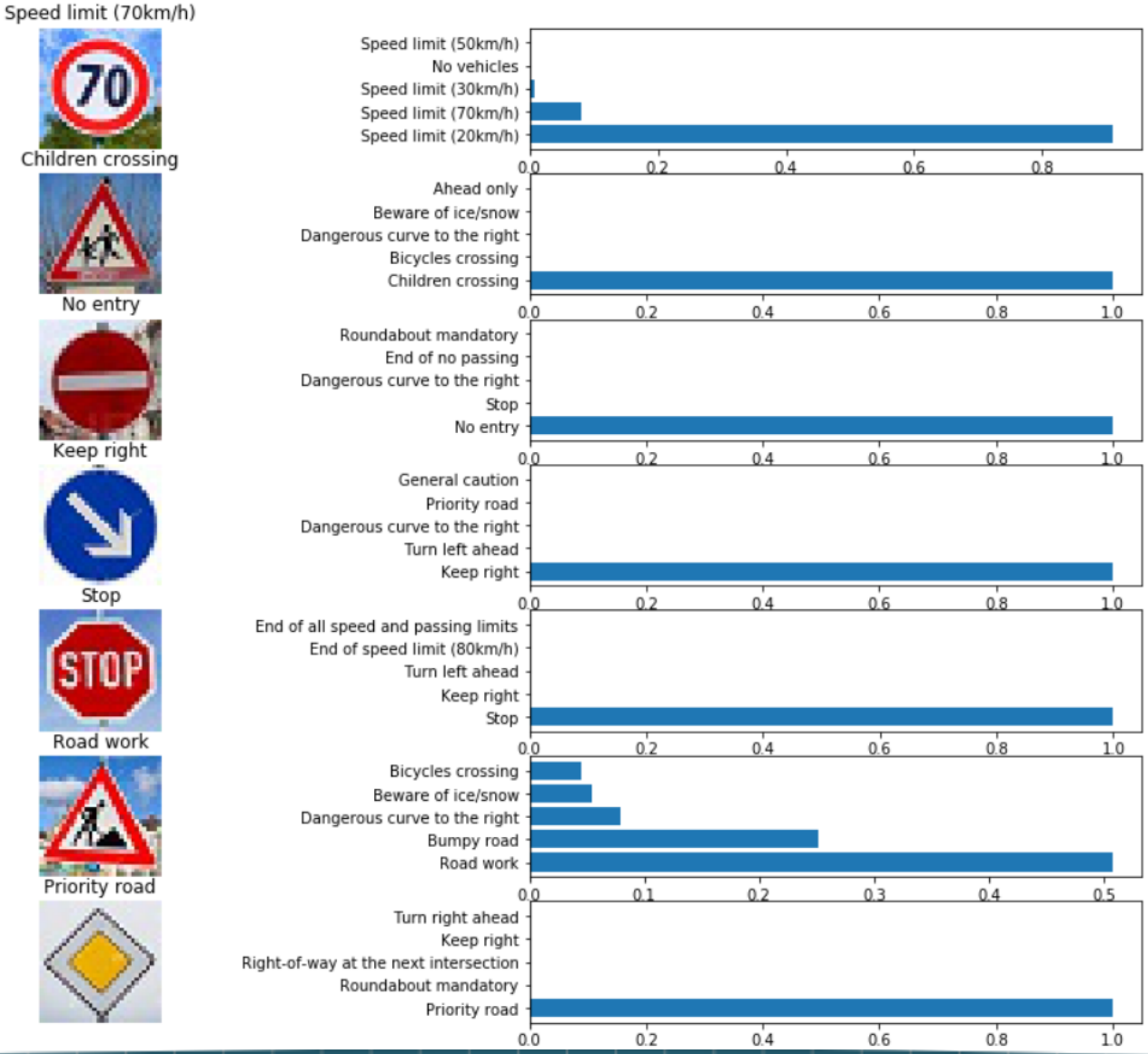



```
EPOCH 1 ... Validation Accuracy = 0.686
EPOCH 2 ... Validation Accuracy = 0.871
EPOCH 3 ... Validation Accuracy = 0.918
EPOCH 4 ... Validation Accuracy = 0.941
EPOCH 5 ... Validation Accuracy = 0.947
EPOCH 6 ... Validation Accuracy = 0.944
EPOCH 7 ... Validation Accuracy = 0.954
EPOCH 8 ... Validation Accuracy = 0.963
EPOCH 9 ... Validation Accuracy = 0.970
EPOCH 10 ... Validation Accuracy = 0.967
EPOCH 11 ... Validation Accuracy = 0.959
EPOCH 12 ... Validation Accuracy = 0.966
EPOCH 13 ... Validation Accuracy = 0.967
EPOCH 14 ... Validation Accuracy = 0.962
EPOCH 15 ... Validation Accuracy = 0.966
EPOCH 16 ... Validation Accuracy = 0.970
EPOCH 17 ... Validation Accuracy = 0.966
EPOCH 18 ... Validation Accuracy = 0.966
EPOCH 19 ... Validation Accuracy = 0.967
EPOCH 20 ... Validation Accuracy = 0.968
```

Test Accuracy = 0.955

Ensaios

- Teste com imagens da internet

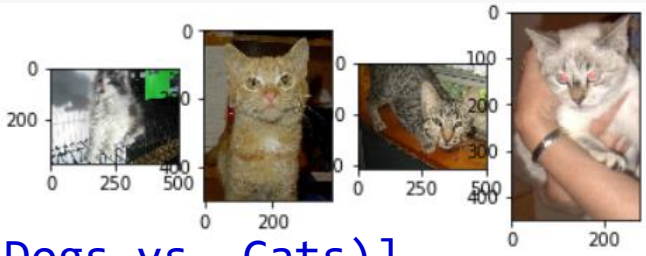


Ensaios – VGG16

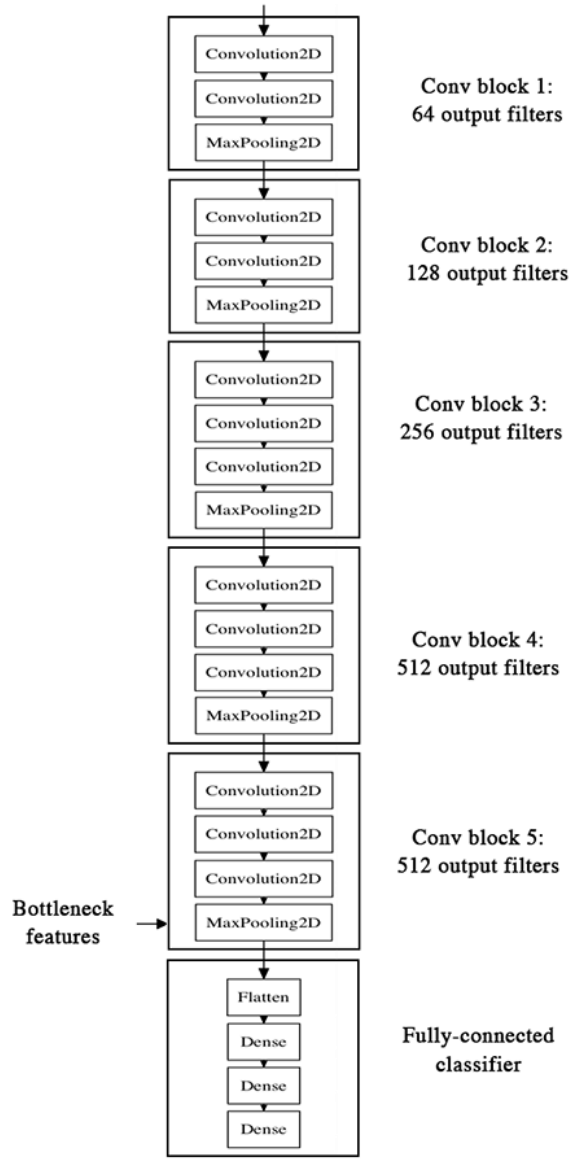
Keras

batch_size = 100
epochs = 100
lr = 0.0001
momentum = 0.9
input_shape = (224,224,3)
train_size = 2000
valid_size = 500
Dropout = 0.5
Data Augmentation:
Rotação, flip

```
# Criação do modelo com só camadas fully-connected
classification_layers = [Flatten(input_shape=train_data.data.shape[1:]),
Dense(256, activation="relu"),
Dropout(0.5),
Dense(2, activation="softmax")]
model = Sequential(classification_layers)
model.compile(loss=losses.categorical_crossentropy, optimizer=keras.optimizers.SGD
 (lr=lr, momentum=momentum), metrics=['accuracy'])
```



Kaggle [(Dogs vs. Cats)]
(https://www.kaggle.com/c/dogs-vs-cats)



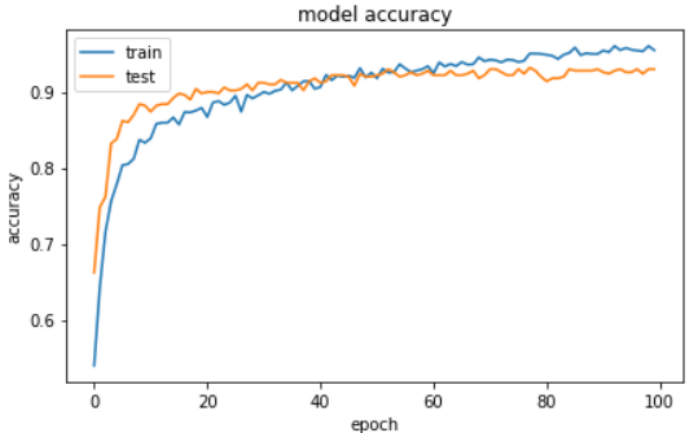
Ensaios – VGG16

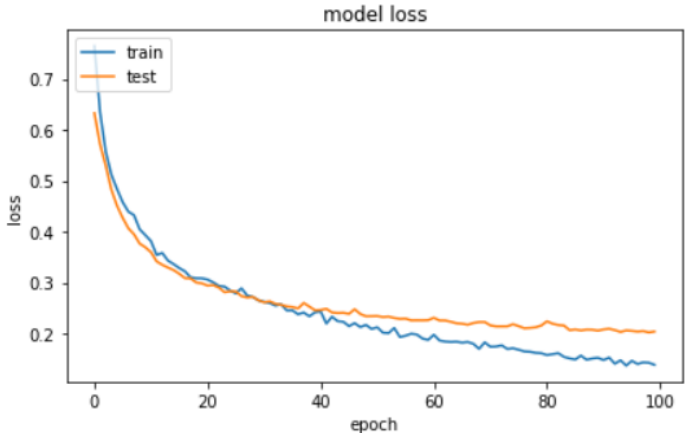
```
Epoch: 98 - loss: 0.147, acc: 0.953:  30%|          | 600/2000 [00:00<00:00, 2416.10it/s]
Epoch: 98 - loss: 0.147, acc: 0.956:  45%|          | 900/2000 [00:00<00:00, 2472.69it/s]
Epoch: 98 - loss: 0.147, acc: 0.958:  60%|          | 1200/2000 [00:00<00:00, 2545.83it/s]
Epoch: 98 - loss: 0.141, acc: 0.962:  75%|          | 1500/2000 [00:00<00:00, 2592.94it/s]
Epoch: 98 - loss: 0.140, acc: 0.962:  90%|          | 1800/2000 [00:00<00:00, 2620.10it/s]
Epoch: 98 - loss: 0.143, acc: 0.961, val_loss: 0.202, val_acc: 0.930: 100%|          | 2000/2000 [00:00<00:00, 1864.12it/s]
Training:  99%|          | 99/100 [01:29<00:01,  1.02s/it]
Epoch: 99:   0%|          | 0/2000 [00:00<?, ?it/s]
Epoch: 99 - loss: 0.159, acc: 0.940:  15%|          | 300/2000 [00:00<00:00, 2759.66it/s]
Epoch: 99 - loss: 0.156, acc: 0.942:  30%|          | 600/2000 [00:00<00:00, 2744.56it/s]
Epoch: 99 - loss: 0.145, acc: 0.949:  45%|          | 900/2000 [00:00<00:00, 2749.06it/s]
Epoch: 99 - loss: 0.143, acc: 0.949:  60%|          | 1200/2000 [00:00<00:00, 2767.42it/s]
Epoch: 99 - loss: 0.139, acc: 0.954:  75%|          | 1500/2000 [00:00<00:00, 2690.88it/s]
Epoch: 99 - loss: 0.141, acc: 0.954:  90%|          | 1800/2000 [00:00<00:00, 2711.15it/s]
Epoch: 99 - loss: 0.139, acc: 0.955, val_loss: 0.204, val_acc: 0.930: 100%|          | 2000/2000 [00:00<00:00, 1912.10it/s]
Training: 100%|          | 100/100 [01:30<00:00,  1.03it/s]
```

Visualização de resultados:

In [31]:
```
from utils import open_object
result = open_object('VGG16_TransferLearning.pkl')
result.show_result()
```

```
Test score: 0.20399780881404878
Test accuracy: 0.9299999990463257
```





Referências Bibliográficas

[1] Jost Tobias Springenberg and Alexey Dosovitskiy and Thomas Brox and Martin A. Riedmiller, 2015: **Striving for Simplicity: The All Convolutional Net**

[2] Transfer Learning – Towards Data Science. [online] Disponível em: https://towardsdatascience.com/transfer-learning-946518f95666 [Acesso em 26 Nov. 2018].

[3] A Gentle Introduction to Transfer Learning for Deep Learning. [online] Disponível em: https://machinelearningmastery.com/transfer-learning-for-deep-learning [Acesso em 26 Nov. 2018].

[4] Transfer learning from pre-trained models – Towards Data Science. [online] Disponível em: https://towardsdatascience.com/transfer-learning-from-pre-trained-models-f2393f124751 [Acesso em 26 Nov. 2018].

[5] Cao, B., Pan, S. J., Zhang, Y., Yeung, D. Y., & Yang, Q. (2010, July). **Adaptive Transfer Learning**. In AAAI (Vol. 2, No. 5, p. 7).

[6] JasonYosinski, JeffClune, YoshuaBengio, and HodLipson, "**How Transferable are Features in Deep Neural Networks?**" In Advances in Neural Information Processing Systems 27 (NIPS '14), NIPS Foundation, 2014.

[7] Soria Olivas, E. (2010). **"Handbook of research on machine learning applications and trends"**. Hershey, Pa.: IGI Global (701 E. Chocolate Avenue, Hershey, Pennsylvania, 17033, USA).

[8] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, **"Gradient-based learning applied to document recognition"**, Proc. IEEE 86(11): 2278–2324, 1998.

[9] Karen Simonyan∗ & Andrew Zisserma, **"VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION"**, ICLR 2015.

Muito Obrigado!!!