

DESMISTIFICANDO O DEEP LEARNING COM TENSORFLOW

Júlio César Batista julio.batista@outlook.com

Vítor Albiero vitor_albiero@outlook.com

May 03, 2017

<https://goo.gl/aYaT7Q>



THE
DEVELOPER'S
CONFERENCE



- Redes neurais foram muito utilizadas até os anos 2000;
- Houve um hiato relacionado as redes neurais com o uso de outros algoritmos (SVM, árvores de decisão);
- Em 2012 um resultado¹ trouxe muita atenção para as redes neurais convolucionais;

Model	Top-1	Top-5
Sparse coding	47.1%	28.2%
SIFT + FVs	45.7%	25.7%
CNN	37.5%	17.0%

¹Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances In Neural Information Processing Systems* (2012).

Entendendo o problema

- Reconhecimento de imagens: Dada a imagem de um objeto (caractère), retorne o objeto (caractère) está nessa imagem;
- Aprendizado supervisionado: Um algoritmo aprende a classificar uma imagem a partir de um conjunto de imagens que foram anotadas para indicar qual objeto está em cada imagem;



Caractère do MNIST²

²Y. Lecun et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* (1998).

Como representar imagens

- Imagens são matrizes onde os seus valores (*pixels*) variam de 0 (preto) a 255 (branco);
- Essas matrizes podem ser transformadas em vetores (*arrays*) através de uma operação chamada *flatten*;



$$\mathbf{x}_{28 \times 28} = \begin{bmatrix} 0 & \cdots & \cdots & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & 255 & 255 & \ddots & \vdots \\ \vdots & 255 & 255 & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & 0 \end{bmatrix}$$

$$\text{flatten}(\mathbf{x}) = [\cdots \ 255 \ 255 \ \cdots \ 255 \ 255 \ \cdots]_{1 \times 784}$$

Um *framework* para transformar imagens em *labels*

- Multiplicando a imagem de entrada (\mathbf{x}) por uma matriz de *weights* (\mathbf{W}) e adicionando um vetor de *bias* (\mathbf{b}) podemos ter um vetor ($\hat{\mathbf{y}}$) de C classes;

$$\hat{\mathbf{y}}_{C \times 1} = \mathbf{W}_{C \times 784} \mathbf{x}_{784 \times 1} + \mathbf{b}_{C \times 1}$$
$$\hat{\mathbf{y}}_{C \times 1} = \begin{bmatrix} -0.76 & \cdots & -0.39 \\ 0.07 & \cdots & 0.91 \\ -0.19 & \cdots & 0.23 \\ -0.42 & \cdots & 0.13 \\ \vdots & \ddots & \vdots \\ -0.82 & \cdots & 0.20 \end{bmatrix}_{C \times 784} \times \begin{bmatrix} -0.44 \\ 0.40 \\ 0.11 \\ \vdots \\ -0.66 \end{bmatrix}_{784 \times 1} + \begin{bmatrix} 1.00 \\ -0.18 \\ 0.81 \\ \vdots \\ -1.00 \end{bmatrix}_{C \times 1}$$

Tornando as saídas "interpretáveis"

- Softmax (σ) transforma um vetor de valores arbitrários em um vetor de probabilidades (distribuição);

$$\sigma(\hat{\mathbf{y}})_c = \frac{\exp(\hat{\mathbf{y}}_c)}{\sum_{k=1}^K \exp(\hat{\mathbf{y}}_k)} \quad \hat{\mathbf{y}} = \begin{bmatrix} -0.05 \\ 0.43 \\ 0.92 \\ 0.92 \\ 0.28 \\ -0.69 \\ 0.17 \\ 2.34 \\ -0.81 \\ 0.63 \end{bmatrix} \quad \sigma(\hat{\mathbf{y}}) = \begin{bmatrix} 0.04 \\ 0.07 \\ 0.11 \\ 0.11 \\ 0.06 \\ 0.02 \\ 0.05 \\ 0.45 \\ 0.02 \\ 0.08 \end{bmatrix}$$

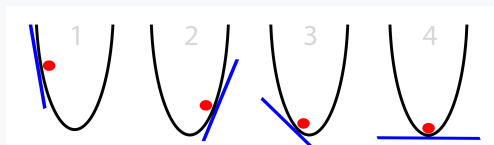
- O "aprendizado" é considerado um problema de otimização de uma função objetivo (*loss/cost*) \mathcal{L} sobre N amostras;

$$\mathbf{Y} = \begin{bmatrix} 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 1.00 & 0.00 & 0.00 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}_{N \times C}$$
$$\hat{\mathbf{Y}} = \begin{bmatrix} 0.04 & 0.07 & 0.11 & 0.11 & 0.06 & 0.02 & 0.05 & 0.45 & 0.02 & 0.08 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}_{N \times C}$$

$$\mathcal{L}(\mathbf{Y}, \hat{\mathbf{Y}}) = -\frac{1}{N} \sum_{n=1}^N \sum_{c=1}^C \mathbf{Y}_{nc} \log(\hat{\mathbf{Y}}_{nc})$$

Aprendizado: otimização

- O método mais comum de otimização é o *Stochastic Gradient Descent* (SGD) com *momentum*;
- Um item importante do SGD é o *learning rate*, responsável por multiplicar os *weights* e *bias* a cada iteração;



Exemplo do SGD³.

³Andrew Trask. *A Neural Network in 13 lines of Python (Part 2 - Gradient Descent)*.

<https://iamtrask.github.io/2015/07/27/python-network-part2/>. Accessed:

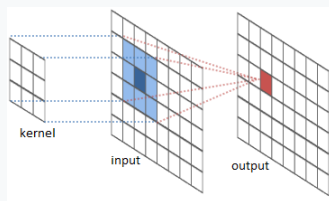
2017-04-27.

DEMO

Entretanto, este *framework*
não aprende *features*.

Convoluções: o processo de filtrar uma imagem

- Classicamente: usado para suavização ou realce de arestas;
- Atualmente: a rede "aprende as melhores" convoluções ajustando \mathbf{W} e \mathbf{b} durante o processo de aprendizado;

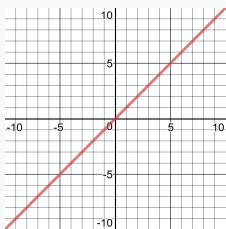


Processo de convolução⁴.

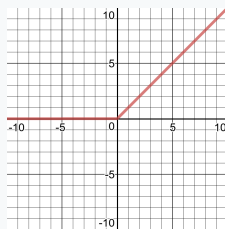
⁴Christopher Olah. *Understanding Convolutions*.

<http://colah.github.io/posts/2014-07-Understanding-Convolutions/>. Accessed: 2017-04-27.

- Basicamente: transforma uma reta em uma curva;
- A mais comum atualmente é a *Rectified Linear Unit* (ReLU)⁵;
- Softmax também é uma função de ativação;



(a) Linear: $\mathbf{y} = \mathbf{W}\mathbf{x} + \mathbf{b}$

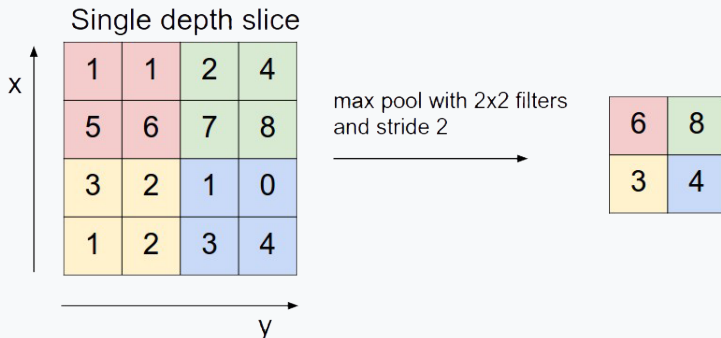


(b) ReLU: $\mathbf{y} = \max(0, \mathbf{W}\mathbf{x} + \mathbf{b})$

⁵Vinod Nair and Geoffrey E Hinton. "Rectified linear units improve restricted boltzmann machines". In: *ICML*. 2010.

Pooling

- Reduz o tamanho (altura, largura) da representação;
- Tipos mais comuns: *max*, *average*;



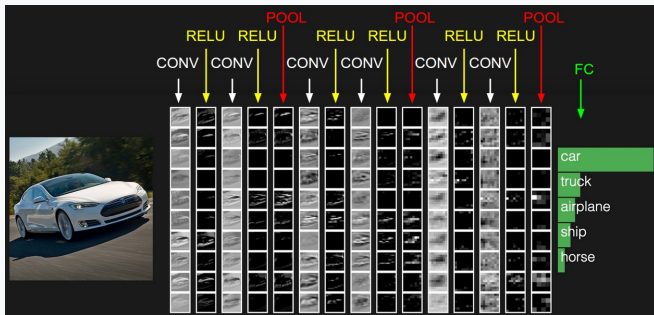
6

⁶CS231n. *Convolutional Neural Networks for Visual Recognition (Pooling Layer)*.

<http://cs231n.github.io/convolutional-networks/>. Accessed: 2017-04-27.

12 / 16

Juntando tudo, enfim as redes neurais convolucionais



Exemplo de uma rede neural convolucional⁷. Note como as camadas de convolução (com relu) e pooling alternam até chegar no final com as totalmente conectadas com softmax.

⁷CS231n. *Convolutional Neural Networks (CNNs / ConvNets)*.

DEMO

Para onde ir e outros frameworks

- Existem vários cursos gratuitos pela internet: [CS231n](#) (Stanford), [CS224n](#) (Stanford), [Deep Learning by Google](#) (Udacity), [Neural Networks for Machine Learning](#) (University of Toronto), [Deep Learning](#) (Oxford), [Neural Networks class](#) (Université de Sherbrooke);
- Livros: [Deep Learning Book](#), [Neural Networks and Deep Learning](#);
- Frameworks: [Keras](#), [Theano](#), [Torch](#), [pyTorch](#), [Caffe](#), [Caffe2](#), [Lasagne](#);
- Mestrado/Doutorado no IMAGO UFPR Research Group (selo de melhor opção garantida);



DESMISTIFICANDO O DEEP LEARNING COM TENSORFLOW

Júlio César Batista julio.batista@outlook.com

Vítor Albiero vitor_albiero@outlook.com

May 03, 2017

<https://goo.gl/aYaT7Q>



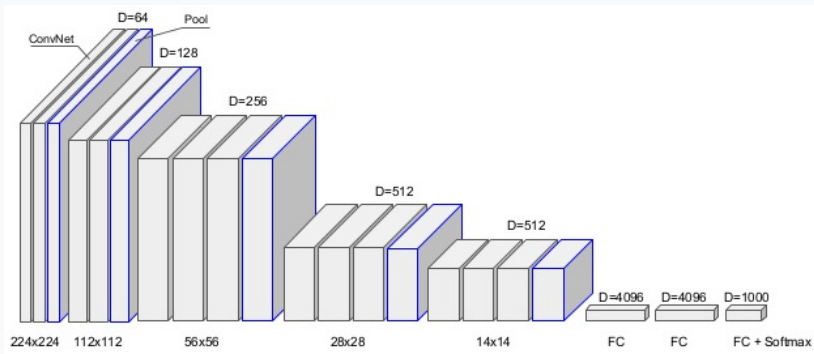
THE
DEVELOPER'S
CONFERENCE



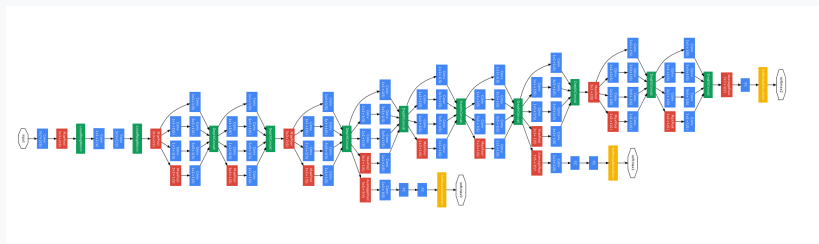
- *Bias* **b**: constante 0
- *Weigths* **W**:
 - Normal (Gaussiano) com $\mu = 0$ e $\sigma = 0.05$
 - Xavier⁸
 - MSRA⁹

⁸Xavier Glorot and Yoshua Bengio. "Understanding the difficulty of training deep feedforward neural networks." In: *Aistats*. 2010.

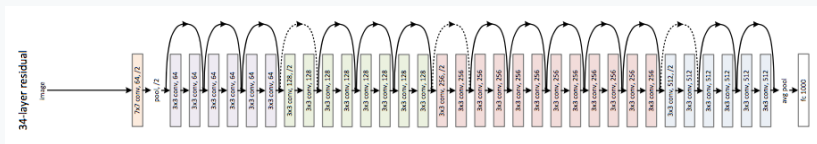
⁹Kaiming He et al. "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification". In: *ICCV*. 2015.



¹⁰Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014).

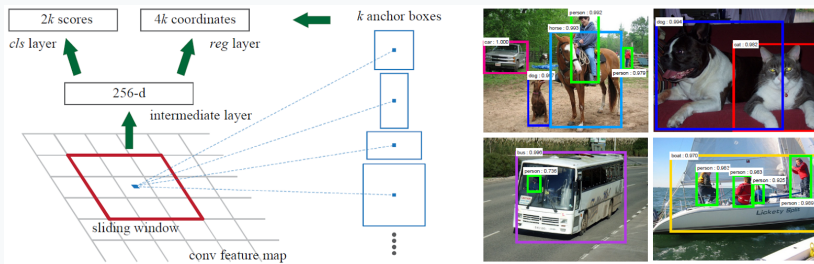


¹¹C. Szegedy et al. "Going deeper with convolutions". In: *IEEE CVPR*. 2015.



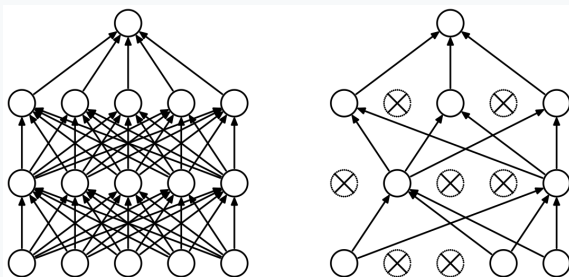
¹²Kaiming He et al. “Deep residual learning for image recognition”. In: *IEEE CVPR*. 2016.

Faster R-CNN¹³



¹³Shaoqing Ren et al. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". In: *IEEE TPAMI* (2016).

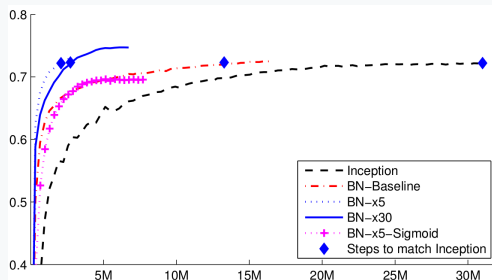
- "Elimina" algumas unidades aleatoriamente durante o treinamento (treina várias arquiteturas diferentes e gera um modelo médio entre as arquiteturas);



¹⁴Nitish Srivastava et al. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *JMLR* (2014).

Batch normalization¹⁵

- Normaliza (média e desvio padrão) as camadas;
- Menos cuidado com inicializações e *learning rate*;



¹⁵Sergey Ioffe and Christian Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: *ICML*. 2015.

- Escolha uma arquitetura (VGG16, GoogLeNet, ResNet, ou crie a sua);
- Treine em uma base genérica e grande (ImageNet¹⁶);
- Altere as saídas para o seu problema e treine na base desejada;

¹⁶Olga Russakovsky et al. "Imagenet large scale visual recognition challenge". In: *IJCV* (2015).

DESMISTIFICANDO O DEEP LEARNING COM TENSORFLOW

Júlio César Batista julio.batista@outlook.com

Vítor Albiero vitor_albiero@outlook.com

May 03, 2017

<https://goo.gl/aYaT7Q>



THE
DEVELOPER'S
CONFERENCE

