

Hands-on Deep Learning - Aula 6

Arquiteturas Modernas e Conclusões

Camila Laranjeira¹, Hugo Oliveira¹², Keiller Nogueira¹²

¹Programa de Pós-Graduação em Ciência da Computação (PPGCC)
Universidade Federal de Minas Gerais

²Interest Group in Pattern Recognition and Earth Observation (PATREO)
Universidade Federal de Minas Gerais

01 de Setembro, 2018



Agenda



1 Introdução

2 Arquiteturas Modernas

3 Feedback

Agenda



1 Introdução

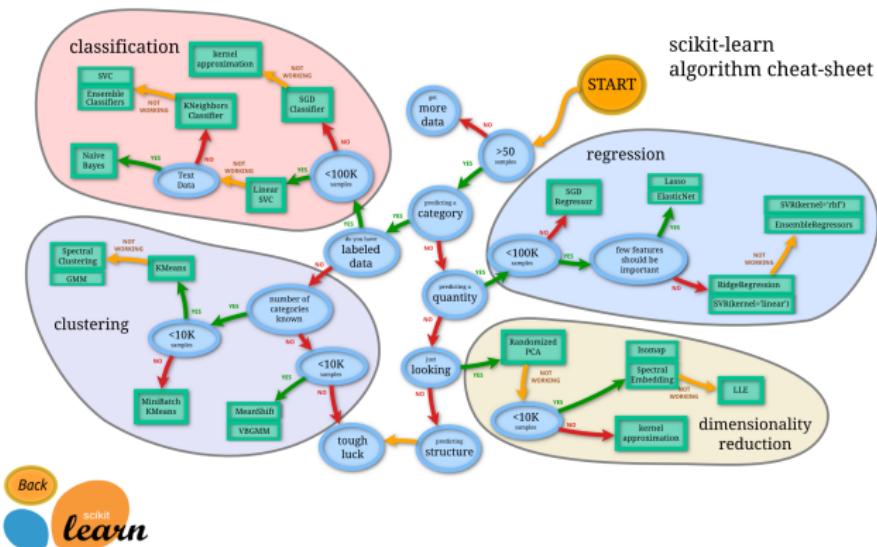
2 Arquiteturas Modernas

3 Feedback



Recapitulando

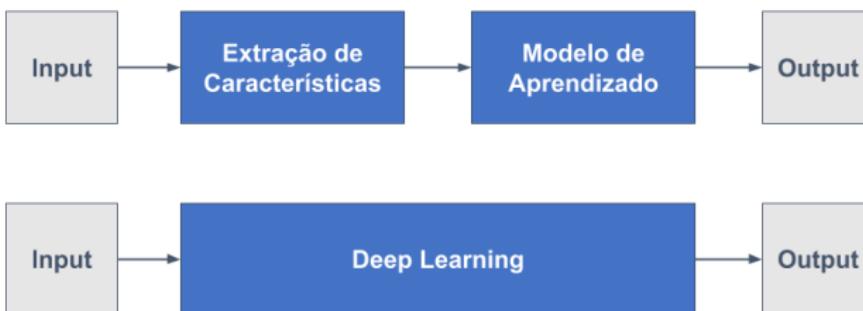
- Algoritmos de Aprendizado de Máquina (Mapa de Métodos)





Recapitulando

- Deep Learning é uma metodologia que simultaneamente aprende as características do dado e treina um modelo de aprendizado.



Recapitulando

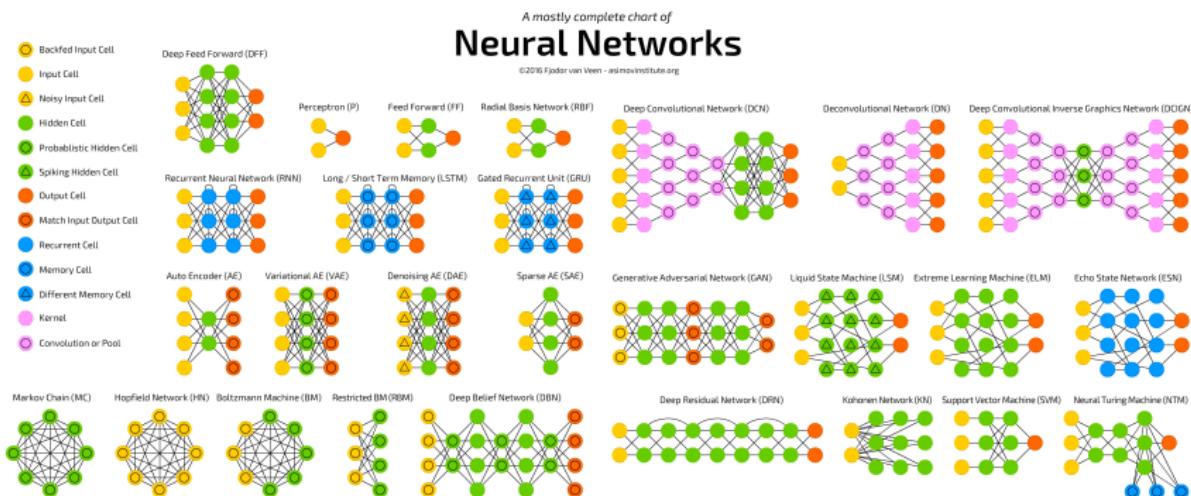


- Teorema de Aproximação Universal
 - "*Uma rede neural feed forward com apenas uma camada (escondida) é suficiente para representar qualquer função, mas a camada pode ser inviavelmente grande e pode falhar em aprender e generalizar corretamente.*"



Recapitulando

- Zoológico de Arquiteturas



Recapitulando



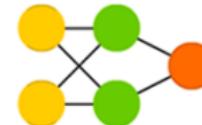
- Redes Neurais ("Vanilla")

Perceptron (P)

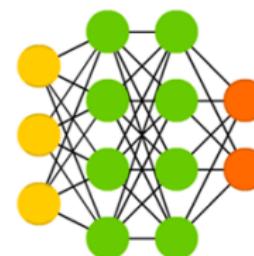
- Input Cell
- Output Cell
- Hidden Cell



Feed Forward (FF)



Deep Feed Forward (DFF)



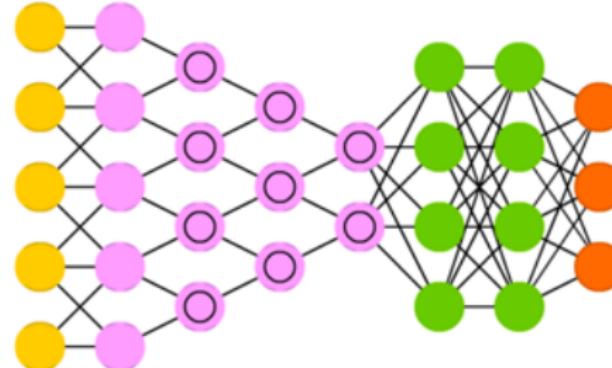
Recapitulando



- Redes Neurais Convolucionais

Deep Convolutional Network (DCN)

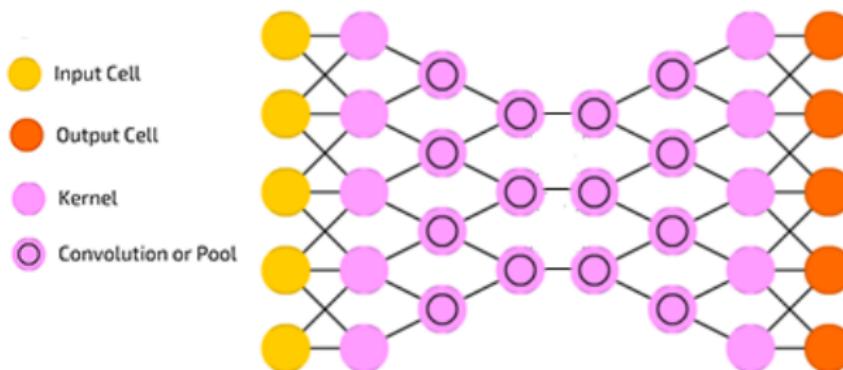
- Input Cell
- Output Cell
- Hidden Cell
- Kernel
- Convolution or Pool





Recapitulando

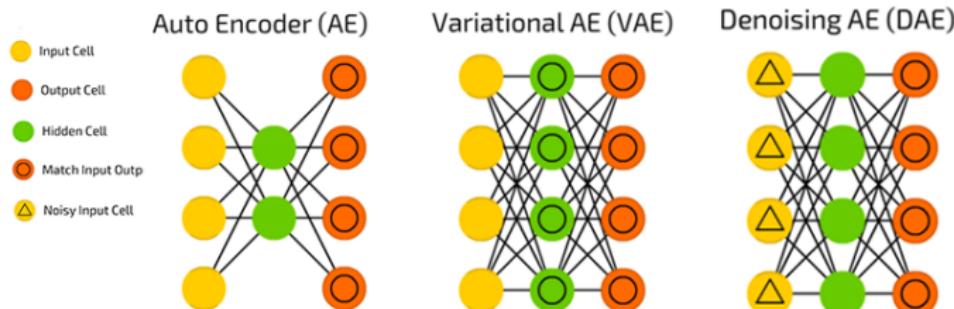
- Fully Convolutional Networks





Recapitulando

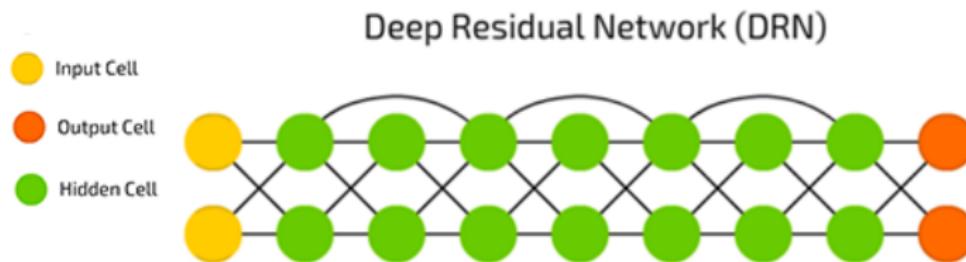
• AutoEncoders



Recapitulando



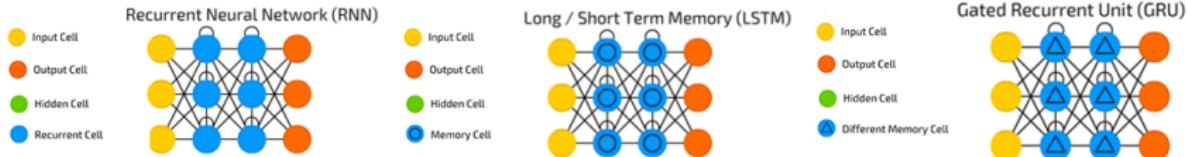
- Residuais



Recapitulando



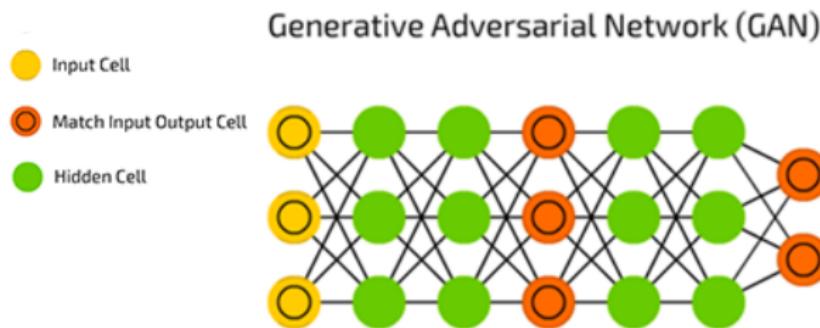
● Redes Recorrentes



Recapitulando



- Generative Adversarial Networks



O que fazer com esse conhecimento?



- Aprofundá-lo e implementar soluções inovadoras.
- Aplicar arquiteturas e modelos existentes no contexto do seu problema.
 - E tem muitas disponíveis gratuitamente!
 - Ou pelo menos artigos explicando a ideia.

Agenda



1 Introdução

2 Arquiteturas Modernas

3 Feedback



Análise de Imagens em Geral

• MobileNets [1]

- Eficiência de Parâmetros
- Adaptável para várias aplicações diferentes
- Convoluçãoes depthwise
- Convoluçãoes 1x1
- Aplicações Mobile

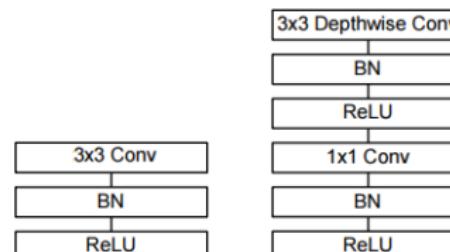




Análise de Imagens em Geral

- MobileNets [1]

- Eficiência de Parâmetros
- Adaptável para várias aplicações diferentes
- Convoluçãoes depthwise
- Convoluçãoes 1x1
- Aplicações Mobile



(a) Módulos convolucionais normais.
(b) Módulos convolucionais de MobileNets.

Análise de Imagens em Geral



- MobileNets [1]
 - Eficiência de Parâmetros
 - Adaptável para várias aplicações diferentes
 - Convoluçãoções depthwise
 - Convoluçãoções 1x1
 - Aplicações Mobile

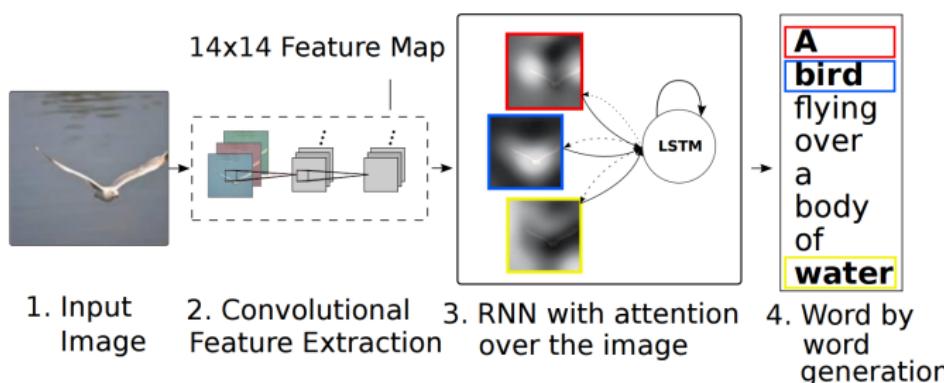
Implementações Base em Pytorch e Tensorflow

- <https://github.com/marvis/pytorch-mobilenet>
- <https://github.com/Zehaos/MobileNet>



Descrição Textual de Cenas

- Attention Networks [2]
 - Camadas Recorrentes com Camadas Convolucionais
 - A mesma imagem é analisada várias vezes em sequência (para cada palavra gerada)
 - As Camadas Recorrentes definem em qual área das imagens as convoluções devem “prestar atenção”

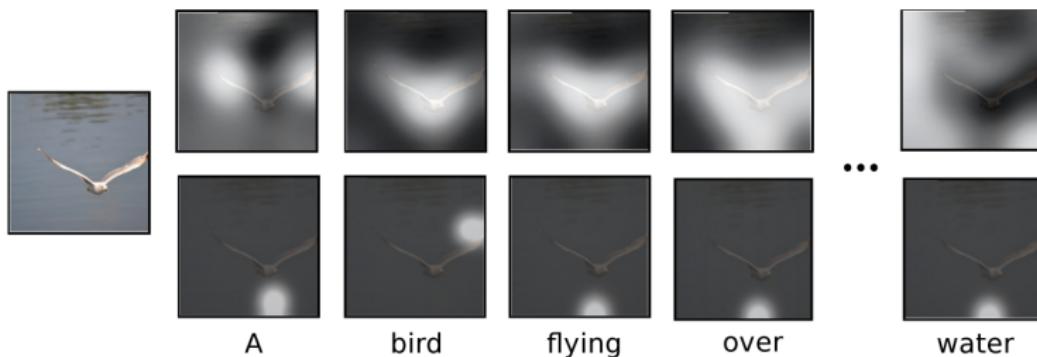




Descrição Textual de Cenas

- Attention Networks [2]

- Camadas Recorrentes com Camadas Convolucionais
 - A mesma imagem é analisada várias vezes em sequência (para cada palavra gerada)
 - As Camadas Recorrentes definem em qual área das imagens as convoluções devem “prestar atenção”





Descrição Textual de Cenas

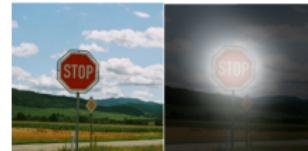
- Attention Networks [2]
 - Camadas Recorrentes com Camadas Convolucionais
 - A mesma imagem é analisada várias vezes em sequência (para cada palavra gerada)
 - As Camadas Recorrentes definem em qual área das imagens as convoluções devem “prestar atenção”



A woman is throwing a frisbee in a park.



A dog is standing on a hardwood floor.



A stop sign is on a road with a mountain in the background.



A little girl sitting on a bed with a teddy bear.



A group of people sitting on a boat in the water.



A giraffe standing in a forest with trees in the background.

Descrição Textual de Cenas



- Attention Networks [2]

- Camadas Recorrentes com Camadas Convolucionais
 - A mesma imagem é analisada várias vezes em sequência (para cada palavra gerada)
 - As Camadas Recorrentes definem em qual área das imagens as convoluções devem “prestar atenção”

Implementação em Pytorch

https://github.com/parksunwoo/show_attend_and_tell_pytorch



Visual Question and Answer

- Memory, Attention and Composition (MAC) Networks [3]
 - Capazes de responder perguntas de **altíssimo nível semântico**
 - Mistura de RNNs com CNNs

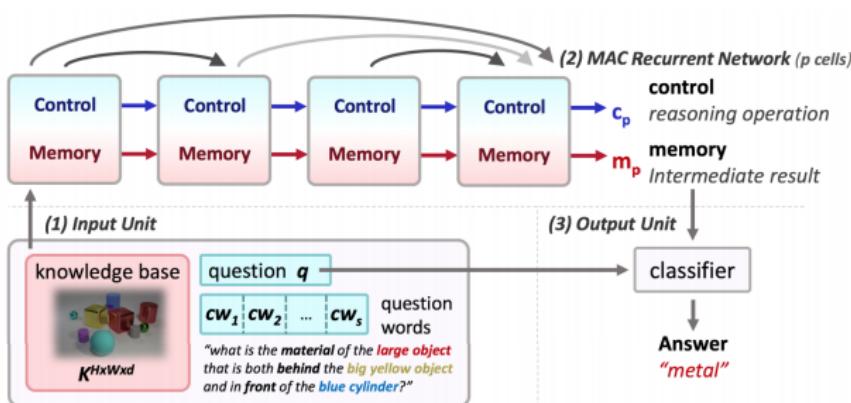


Q: Do **the block** in front of **the tiny yellow cylinder** and **the tiny thing** that is to the right of **the large green shiny object** have **the same color**? **A:** No



Visual Question and Answer

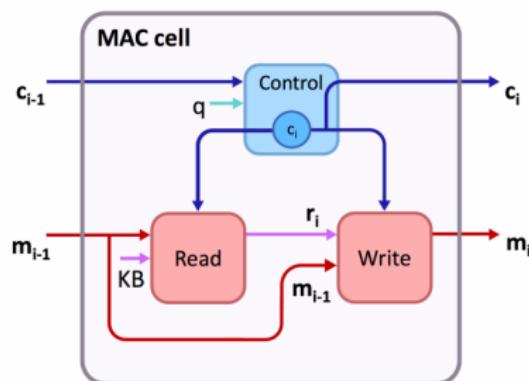
- Memory, Attention and Composition (MAC) Networks [3]
 - Capazes de responder perguntas de **altíssimo nível semântico**
 - Mistura de RNNs com CNNs





Visual Question and Answer

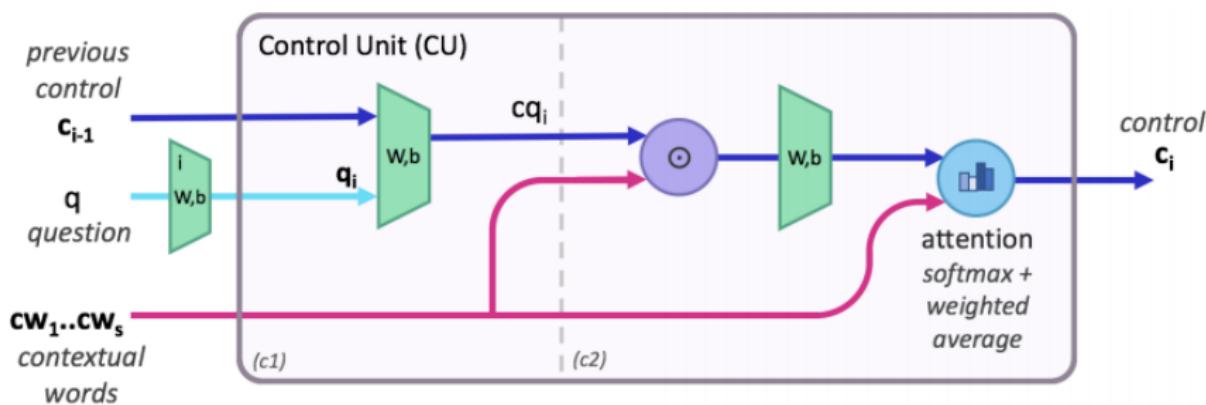
- Memory, Attention and Composition (MAC) Networks [3]
 - Capazes de responder perguntas de **altíssimo nível semântico**
 - Mistura de RNNs com CNNs





Visual Question and Answer

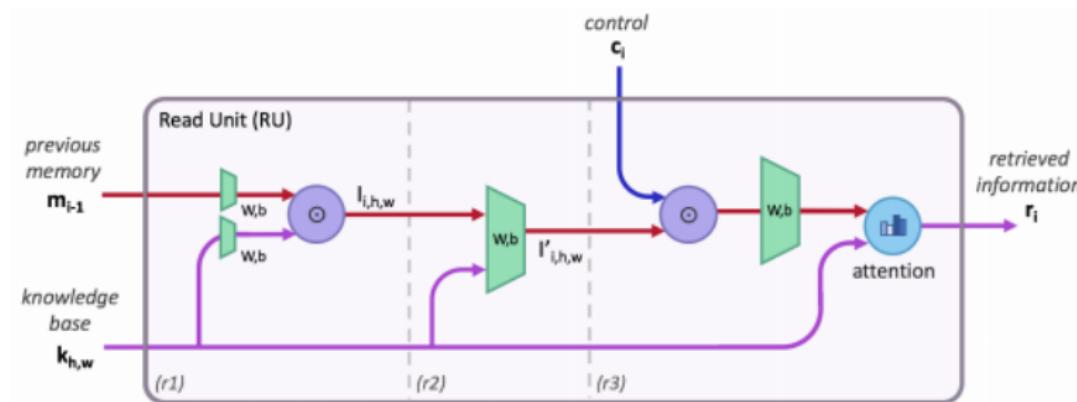
- Memory, Attention and Composition (MAC) Networks [3]
 - Capazes de responder perguntas de **altíssimo nível semântico**
 - Mistura de RNNs com CNNs



Visual Question and Answer



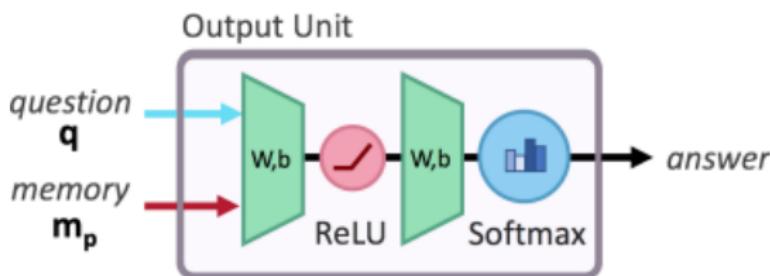
- Memory, Attention and Composition (MAC) Networks [3]
 - Capazes de responder perguntas de **altíssimo nível semântico**
 - Mistura de RNNs com CNNs





Visual Question and Answer

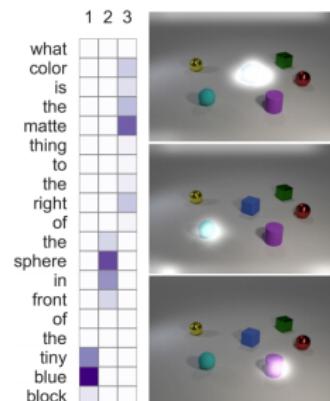
- Memory, Attention and Composition (MAC) Networks [3]
 - Capazes de responder perguntas de **altíssimo nível semântico**
 - Mistura de RNNs com CNNs



Visual Question and Answer



- Memory, Attention and Composition (MAC) Networks [3]
 - Capazes de responder perguntas de **altíssimo nível semântico**
 - Mistura de RNNs com CNNs



Visual Question and Answer



- Memory, Attention and Composition (MAC) Networks [3]
 - Capazes de responder perguntas de **altíssimo nível semântico**
 - Mistura de RNNs com CNNs

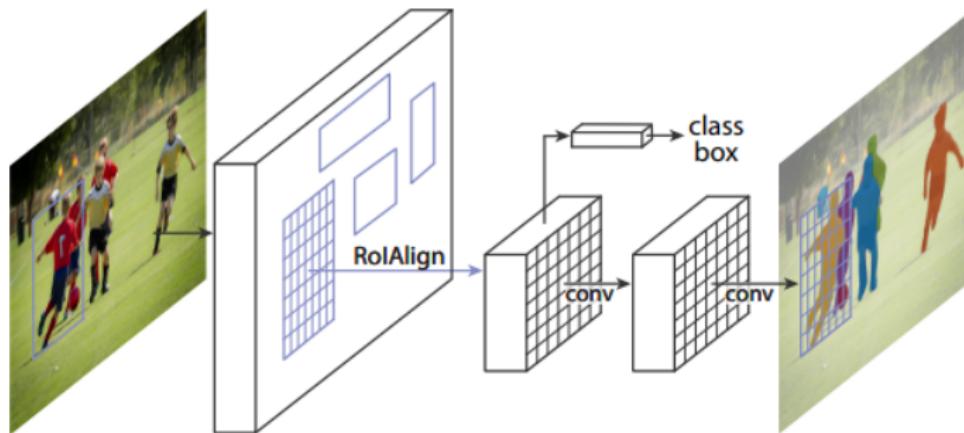
Implementação em Pytorch

<https://github.com/rosinality/mac-network-pytorch>



Segmentação e Detecção

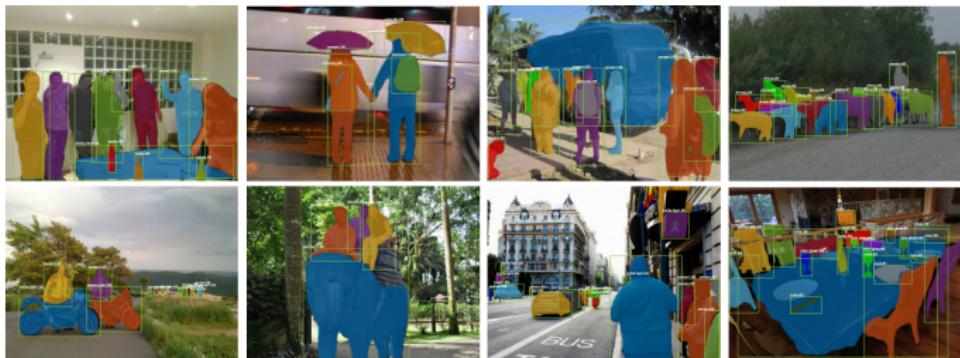
- Mask R-CNNs [4]
 - Detecção e Segmentação ao mesmo tempo
 - Rotulação Densa com as **limitações de tempo** de problemas de Detecção
 - Usa como base ResNets





Segmentação e Detecção

- Mask R-CNNs [4]
 - Detecção e Segmentação ao mesmo tempo
 - Rotulação Densa com as **limitações de tempo** de problemas de Detecção
 - Usa como base ResNets



Segmentação e Detecção



- Mask R-CNNs [4]
 - Detecção e Segmentação ao mesmo tempo
 - Rotulação Densa com as **limitações de tempo** de problemas de Detecção
 - Usa como base ResNets

Implementação Base em Keras

https://github.com/matterport/Mask_RCNN



Remoção de Ruído

- Noise2Noise [5]

- Treinamento feito **apenas** com datasets infectados por ruído
 - Algumas suposições sobre a distribuição do ruído nos dados
- Poucos detalhes de implementação no artigo

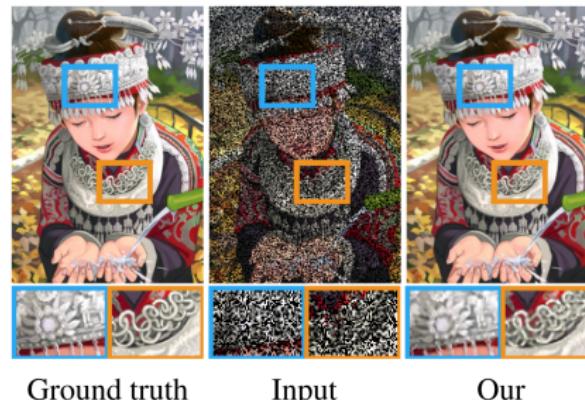


Figura: Ruído Bernoulli



Remoção de Ruído

- Noise2Noise [5]

- Treinamento feito **apenas** com datasets infectados por ruído
 - Algumas suposições sobre a distribuição do ruído nos dados
- Poucos detalhes de implementação no artigo

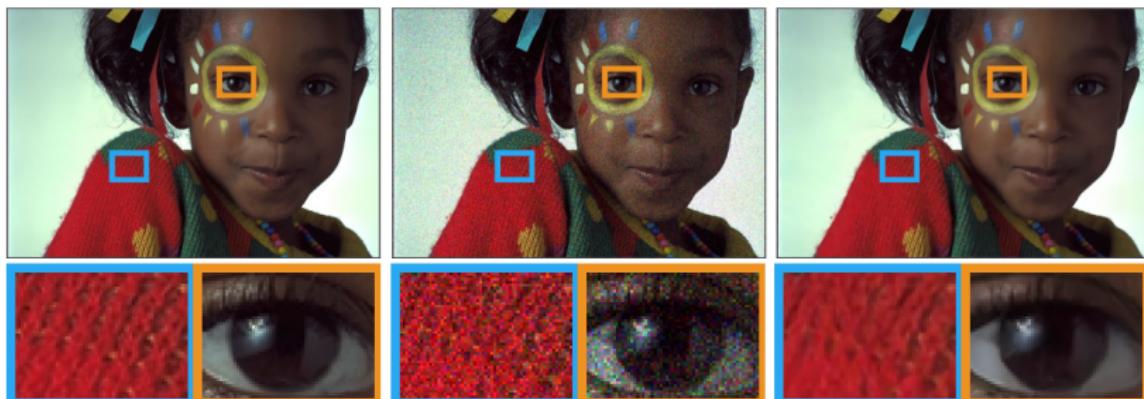


Figura: Ruído Poisson



Remoção de Ruído

- Noise2Noise [5]

- Treinamento feito **apenas** com datasets infectados por ruído
 - Algumas suposições sobre a distribuição do ruído nos dados
- Poucos detalhes de implementação no artigo

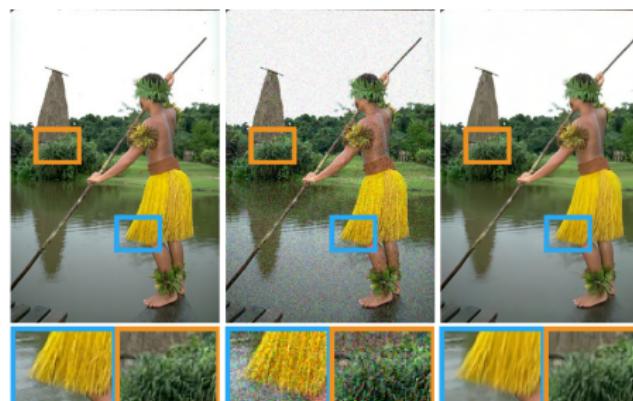


Figura: Ruído Gaussiano



Remoção de Ruído

- Noise2Noise [5]

- Treinamento feito **apenas** com datasets infectados por ruído
 - Algumas suposições sobre a distribuição do ruído nos dados
- Poucos detalhes de implementação no artigo



Figura: Ruído Aleatório



Remoção de Ruído

- Noise2Noise [5]

- Treinamento feito **apenas** com datasets infectados por ruído
 - Algumas suposições sobre a distribuição do ruído nos dados
- Poucos detalhes de implementação no artigo



Figura: Sobreposição de textos



Remoção de Ruído

- Noise2Noise [5]
 - Treinamento feito **apenas** com datasets infectados por ruído
 - Algumas suposições sobre a distribuição do ruído nos dados
 - Poucos detalhes de implementação no artigo

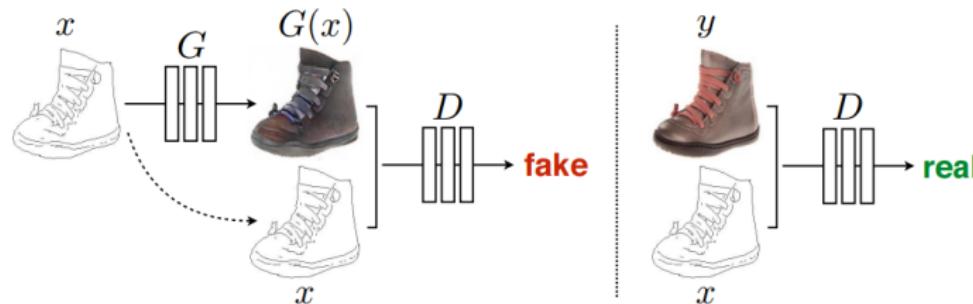
Implementação Base em Keras

<https://github.com/yu4u/noise2noise>



Tradução de Imagens

- pix2pix [6]
 - Imagens **pareadas** nos dois domínios
 - Loss L1 entre a imagem traduzida e a original do outro domínio
 - Generativa agora é uma arquitetura Encoder-Decoder (U-nets, SegNets, Autoencoders...)
 - Discriminativa agora está **condicionada** à imagem de entrada



Tradução de Imagens



- pix2pix [6]
 - Imagens **pareadas** nos dois domínios
 - Loss L1 entre a imagem traduzida e a original do outro domínio
 - Generativa agora é uma arquitetura Encoder-Decoder (U-nets, SegNets, Autoencoders...)
 - Discriminativa agora está **condicionada** à imagem de entrada

Implementações Base em Torch, Pytorch, Tensorflow...

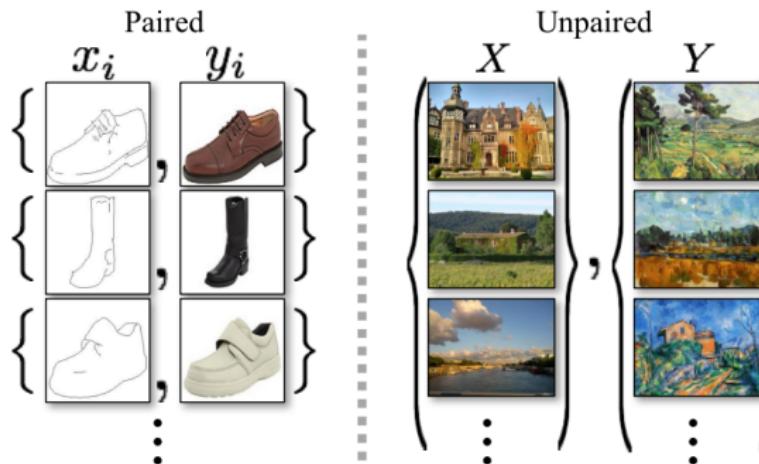
<https://phillipi.github.io/pix2pix/> Demo:

<https://affinelayer.com/pixsrv/>



Tradução de Imagens

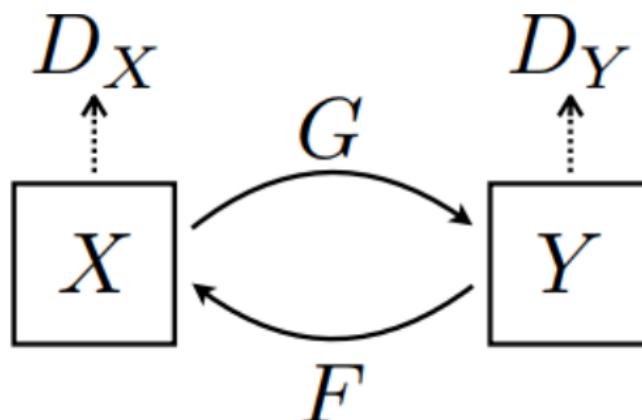
- O pareamento das imagens limita muito as aplicações práticas dessas redes.





Tradução de Imagens

- Cycle GANs [7]
 - Imagens **não pareadas** dos dois domínios
 - Ideia de **Cycle Consistency**
 - Loss é obtida ao **comparar a imagem com ela mesma**

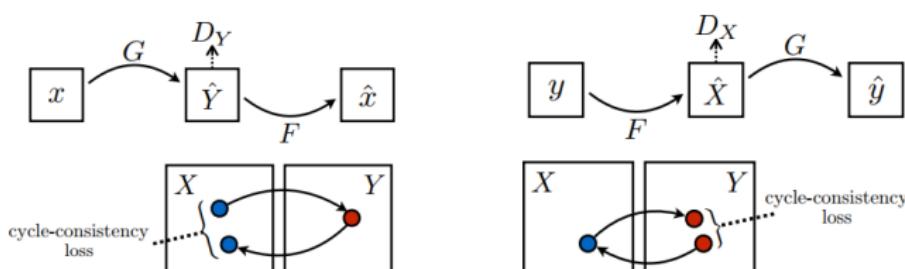


Tradução de Imagens



- Cycle GANs [7]

- Imagens não pareadas dos dois domínios
- Ideia de Cycle Consistency
 - Loss é obtida ao comparar a imagem com ela mesma

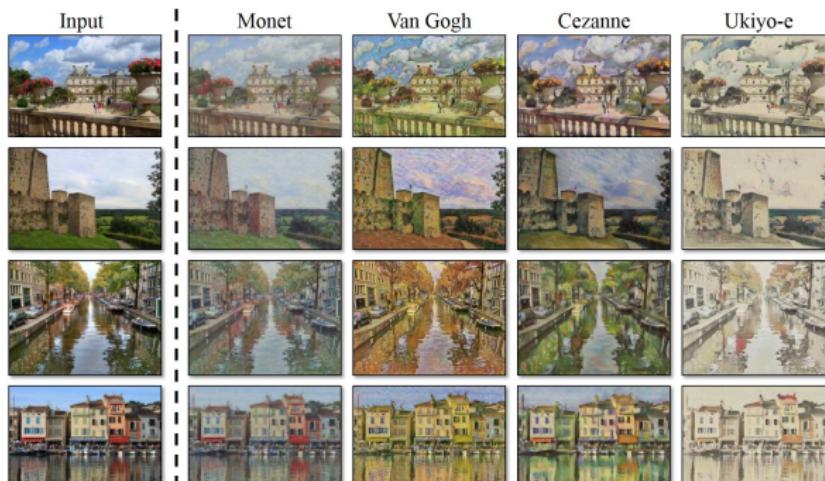




Tradução de Imagens

- Cycle GANs [7]

- Imagens não pareadas dos dois domínios
- Ideia de Cycle Consistency
- Loss é obtida ao comparar a imagem com ela mesma

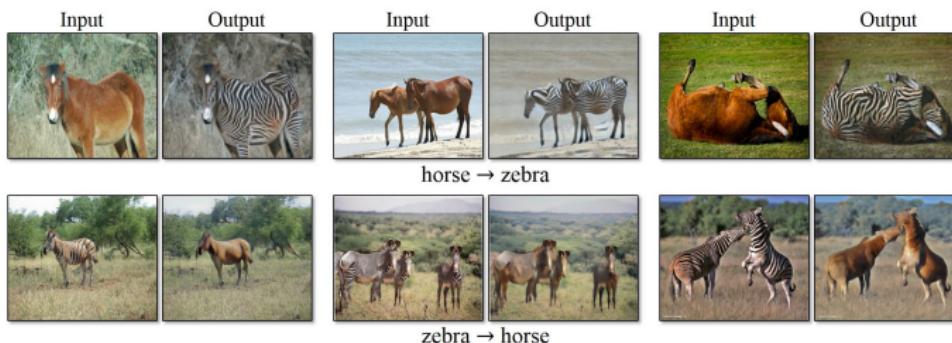




Tradução de Imagens

- Cycle GANs [7]

- Imagens não pareadas dos dois domínios
- Ideia de Cycle Consistency
 - Loss é obtida ao comparar a imagem com ela mesma



Tradução de Imagens



- Cycle GANs [7]
 - Imagens não pareadas dos dois domínios
 - Ideia de Cycle Consistency
 - Loss é obtida ao comparar a imagem com ela mesma

Implementação Base em Torch, Pytorch, Tensorflow...

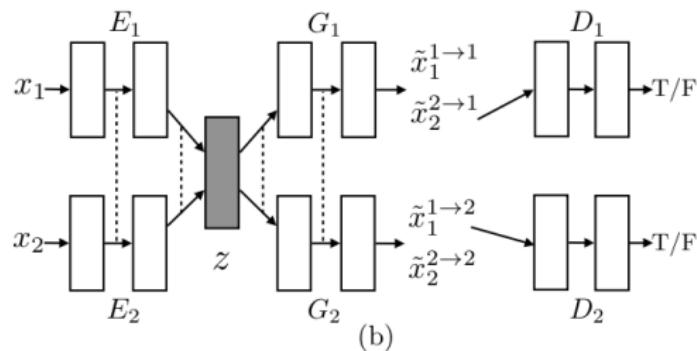
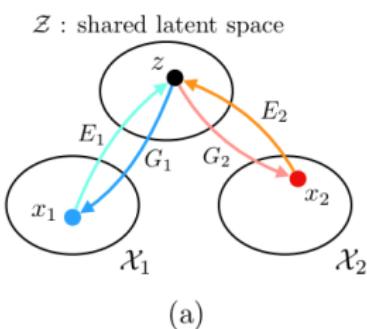
<https://github.com/junyanz/CycleGAN>

Tradução de Imagens



- UNIT [8]

- Imagens **não pareadas** dos dois domínios
- Espaço latente compartilhado entre os domínios





Tradução de Imagens

- UNIT [8]

- Imagens **não pareadas** dos dois domínios
- Espaço latente compartilhado entre os domínios





Tradução de Imagens

- UNIT [8]

- Imagens **não pareadas** dos dois domínios
- Espaço latente compartilhado entre os domínios

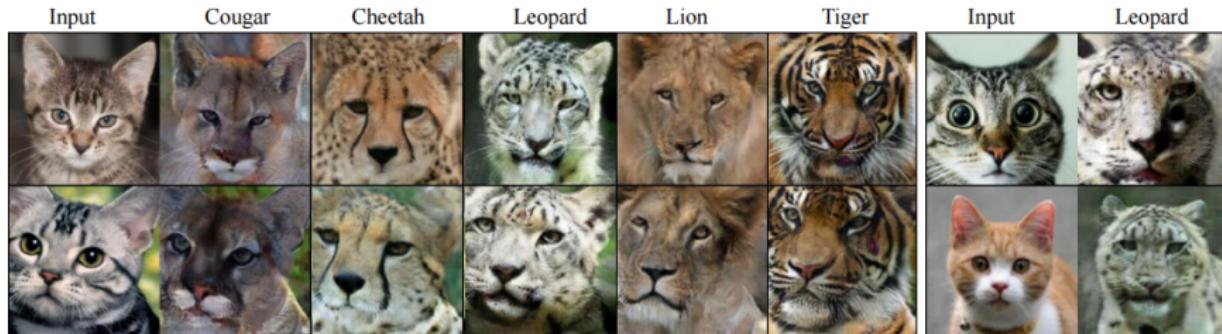




Tradução de Imagens

- UNIT [8]

- Imagens **não pareadas** dos dois domínios
- Espaço latente compartilhado entre os domínios





Tradução de Imagens

- UNIT [8]

- Imagens **não pareadas** dos dois domínios
- Espaço latente compartilhado entre os domínios



Tradução de Imagens



- UNIT [8]
 - Imagens **não pareadas** dos dois domínios
 - Espaço latente compartilhado entre os domínios

Implementações Base em Pytorch

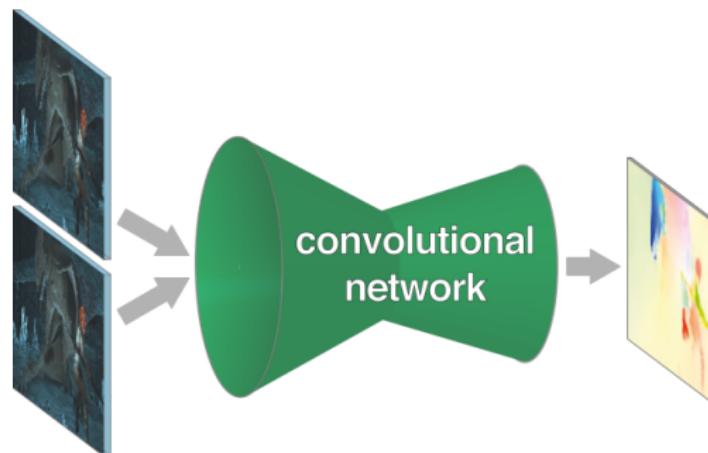
<https://github.com/mingyuliutw/UNIT>



Processamento de Vídeo

- FlowNet [9]

- Aprende a movimentação dos objetos nos vídeos
- Forma Deep de computar Optical Flow entre frames
- Consegue rodar entre 5 e 10 fps

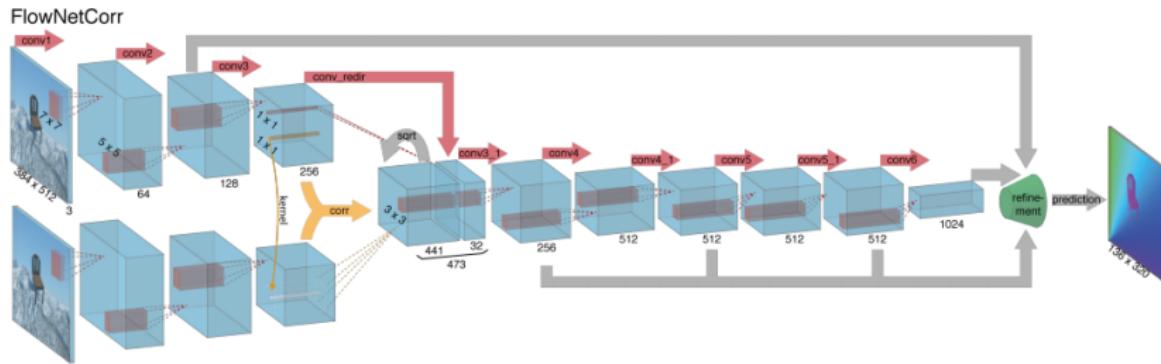




Processamento de Vídeo

- FlowNet [9]

- Aprende a movimentação dos objetos nos vídeos
- Forma Deep de computar Optical Flow entre frames
- Consegue rodar entre 5 e 10 fps





Processamento de Vídeo

- FlowNet [9]

- Aprende a movimentação dos objetos nos vídeos
- Forma Deep de computar Optical Flow entre frames
- Consegue rodar entre 5 e 10 fps

Ground truth



FlowNetS



FlowNetS+v



Processamento de Vídeo



- FlowNet [9]
 - Aprende a movimentação dos objetos nos vídeos
 - Forma Deep de computar Optical Flow entre frames
 - Consegue rodar entre 5 e 10 fps

Implementação Base em Pytorch

<https://github.com/ClementPinard/FlowNetPytorch>

Processamento de Vídeo



- Video-to-Video Synthesis [10]
 - Samples precisam ter coerência temporal
 - Loss Adversarial que força uma correlação temporal correta entre os frames

Implementação Base em Pytorch

<https://github.com/NVIDIA/vid2vid>

Deep Learning sem NNs



Deep Learning sem NNs

Vimos durante o curso todo arquiteturas diferentes de Redes Neurais Artificiais adaptadas para vários tipos de **dados não-estruturados** diferentes. Fica a pergunta então: **Deep Learning equivale a NNs com várias camadas?**



Deep Learning sem NNs

Deep Learning sem NNs

Deep Learning é, na verdade, um **sinônimo de Feature Learning** (ou seja, do aprendizado de características) feito sobre dados não-estruturados. Deep Learning é uma estratégia de extração de características mais robustas que **features hand-crafted**, como HOG, GLCM, Bag of Words...



Deep Learning sem NNs

Deep Learning sem NNs

Deep Learning usando NNs se popularizou porque conseguimos **paralelizar massivamente usando GPUs** o aprendizado usando NNs (as passagens do forward e do backward). O tipo de paralelismo feito em GPUs não é universalmente mais rápido que em CPUs. Dessa forma, **não há versões da maioria dos algoritmos de Aprendizado de Máquina otimizados para rodarem em GPU.**

Deep Learning sem NNs

Porém...



Deep Learning sem NNs

Deep Learning sem NNs

Deep Learning usando NNs se popularizou porque conseguimos **paralelizar massivamente usando GPUs** o aprendizado usando NNs (as passagens do forward e do backward). O tipo de paralelismo feito em GPUs não é universalmente mais rápido que em CPUs. Dessa forma, **não há versões da maioria dos algoritmos de Aprendizado de Máquina otimizados para rodarem em GPU.**

Deep Learning sem NNs

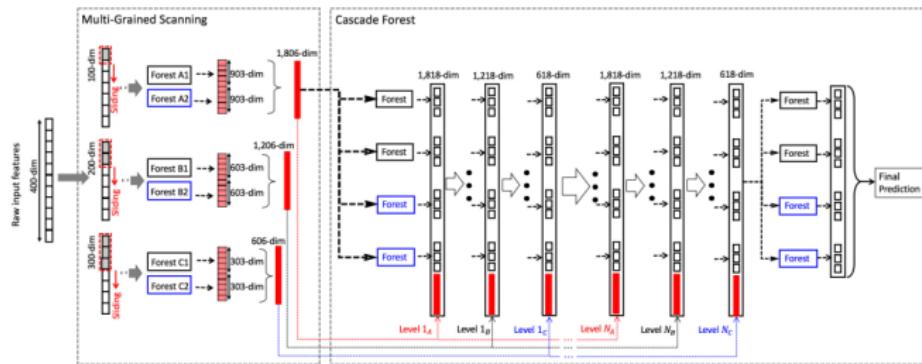
Porém...



Deep Learning sem NNs

- gcForest [11]

- Dados não-estruturados
- Features Deep sem NNs
 - Uso de Árvores de Decisão em camadas
- Futuro do Deep Learning?





Deep Learning sem NNs

- gcForest [11]
 - Dados não-estruturados
 - Features Deep sem NNs
 - Uso de Árvores de Decisão em camadas
 - **Futuro do Deep Learning?**

Table 2: Comparison of test accuracy on MNIST

gcForest	99.26%
LeNet-5	99.05%
Deep Belief Net	98.75% [Hinton <i>et al.</i> , 2006]
SVM (rbf kernel)	98.60%
Random Forest	96.80%



Deep Learning sem NNs

- gcForest [11]
 - Dados não-estruturados
 - Features Deep sem NNs
 - Uso de Árvores de Decisão em camadas
 - Futuro do Deep Learning?

Table 3: Comparison of test accuracy on ORL

	5 image	7 images	9 images
gcForest	91.00%	96.67%	97.50%
Random Forest	91.00%	93.33%	95.00%
CNN	86.50%	91.67%	95.00%
SVM (rbf kernel)	80.50%	82.50%	85.00%
<i>k</i> NN	76.00%	83.33%	92.50%

Deep Learning sem NNs



- gcForest [11]
 - Dados não-estruturados
 - Features Deep sem NNs
 - Uso de Árvores de Decisão em camadas
 - **Futuro do Deep Learning?**

Table 5: Comparison of test accuracy on sEMG data

gcForest	71.30%
LSTM	45.37%
MLP	38.52%
Random Forest	29.62%
SVM (rbf kernel)	29.62%
Logistic Regression	23.33%



Deep Learning sem NNs

- gcForest [11]
 - Dados não-estruturados
 - Features Deep sem NNs
 - Uso de Árvores de Decisão em camadas
 - **Futuro do Deep Learning?**

Table 6: Comparison of test accuracy on IMDB

gcForest	89.16%
CNN	89.02% [Kim, 2014]
MLP	88.04%
Logistic Regression	88.62%
SVM (linear kernel)	87.56%
Random Forest	85.32%

Deep Learning sem NNs



- gcForest [11]
 - Dados não-estruturados
 - Features Deep sem NNs
 - Uso de Árvores de Decisão em camadas
 - Futuro do Deep Learning?

Implementação Base

<https://github.com/kingfengji/gcForest>

Agenda



1 Introdução

2 Arquiteturas Modernas

3 Feedback

Formulário de satisfação



- Preencha o formulário de satisfação com as suas opiniões sobre o curso através do link:

<https://goo.gl/jxSBbx>

Desafio Final



- Que comecem as apresentações!

References

- [1] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. [arXiv preprint arXiv:1704.04861](https://arxiv.org/abs/1704.04861), 2017.
- [2] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In [International conference on machine learning](#), pages 2048–2057, 2015.
- [3] Drew A Hudson and Christopher D Manning. Compositional attention networks for machine reasoning. [arXiv preprint arXiv:1803.03067](https://arxiv.org/abs/1803.03067), 2018.
- [4] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In [Computer Vision \(ICCV\), 2017 IEEE International Conference on](#), pages 2980–2988. IEEE, 2017.
- [5] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. Noise2noise: Learning image restoration without clean data. [arXiv preprint arXiv:1803.04189](https://arxiv.org/abs/1803.04189), 2018.
- [6] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. [arXiv preprint](https://arxiv.org/abs/1611.07004), 2017.
- [7] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. [arXiv preprint arXiv:1703.10593](https://arxiv.org/abs/1703.10593), 2017.
- [8] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In [Advances in Neural Information Processing Systems](#), pages 700–708, 2017.
- [9] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In [Proceedings of the IEEE International Conference on Computer Vision](#), pages 2758–2766, 2015.



└ References

- [10] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro.
Video-to-video synthesis.
[arXiv preprint arXiv:1808.06601](https://arxiv.org/abs/1808.06601), 2018.
- [11] Zhi-Hua Zhou and Ji Feng.
Deep forest: Towards an alternative to deep neural networks.
[arXiv preprint arXiv:1702.08835](https://arxiv.org/abs/1702.08835), 2017.

