



INF-0618

Tópicos em Aprendizado de Máquina II

Aula 5 – Batch normalization / Transfer Learning

Profa. Fernanda Andaló

2018

Instituto de Computação - Unicamp

Batch normalization

Transfer Learning

CPU vs. GPU

Batch normalization

Batch normalization

Técnica utilizada para normalizar os inputs de cada camada da rede neural pela média e variância, a fim de aumentar a estabilidade do treinamento.

BATCH + NORMALIZATION

Batch normalization

Técnica utilizada para normalizar os inputs de cada camada da rede neural pela média e variância, a fim de aumentar a estabilidade do treinamento.

BATCH + NORMALIZATION

Batch normalization

Técnica utilizada para normalizar os inputs de cada camada da rede neural pela média e variância, a fim de aumentar a estabilidade do treinamento.

BATCH + NORMALIZATION

Batch normalization

O que é **batch**?

Treinando sem batch – Stochastic Gradient Descent

Dados de
treinamento

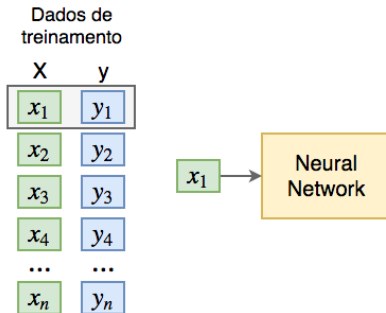
X	y
x_1	y_1
x_2	y_2
x_3	y_3
x_4	y_4
...	...
x_n	y_n

Neural
Network

Batch normalization

O que é **batch**?

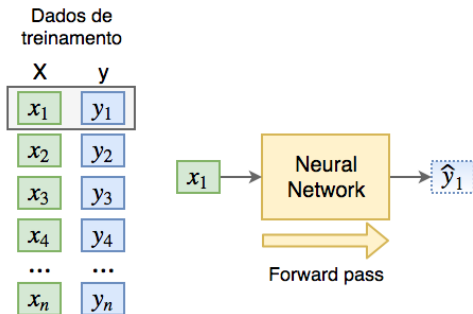
Treinando sem batch – Stochastic Gradient Descent



Batch normalization

O que é **batch**?

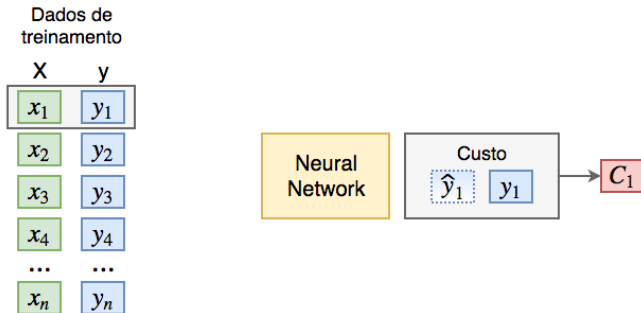
Treinando sem batch – Stochastic Gradient Descent



Batch normalization

O que é **batch**?

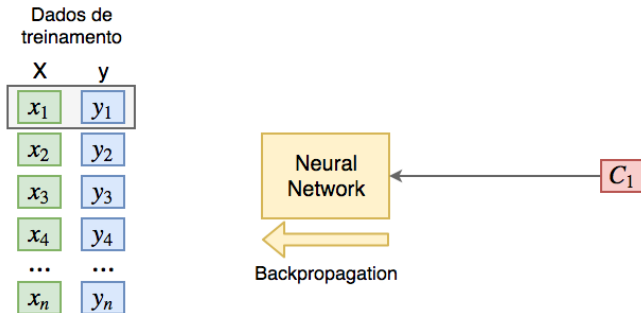
Treinando sem batch – Stochastic Gradient Descent



Batch normalization

O que é **batch**?

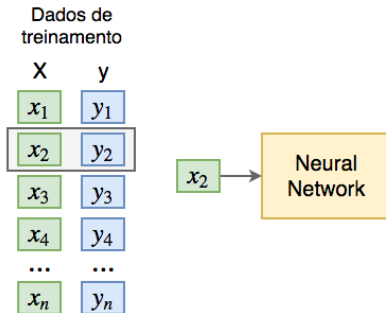
Treinando sem batch – Stochastic Gradient Descent



Batch normalization

O que é **batch**?

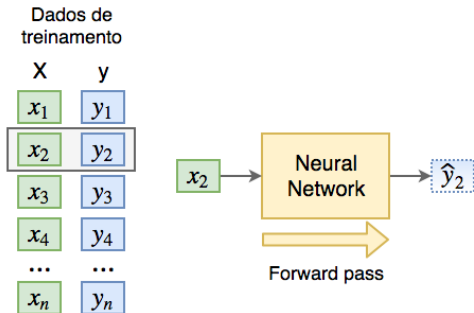
Treinando sem batch – Stochastic Gradient Descent



Batch normalization

O que é **batch**?

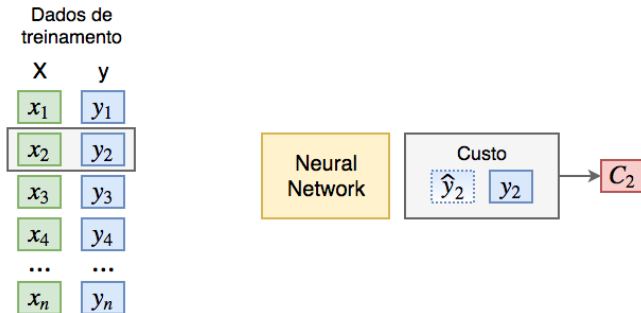
Treinando sem batch – Stochastic Gradient Descent



Batch normalization

O que é **batch**?

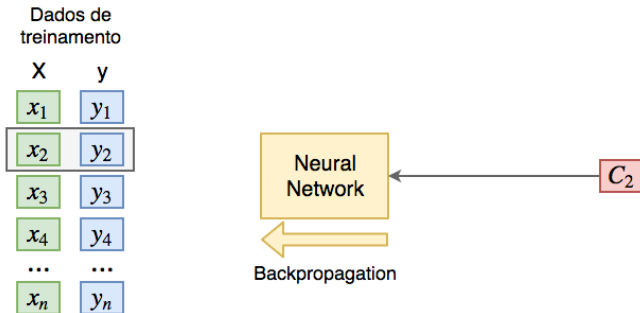
Treinando sem batch – Stochastic Gradient Descent



Batch normalization

O que é **batch**?

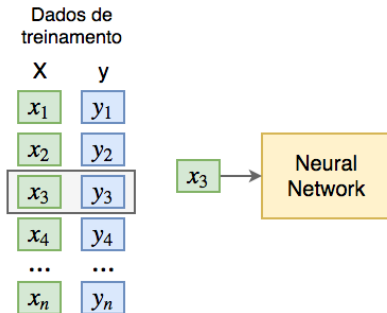
Treinando sem batch – Stochastic Gradient Descent



Batch normalization

O que é **batch**?

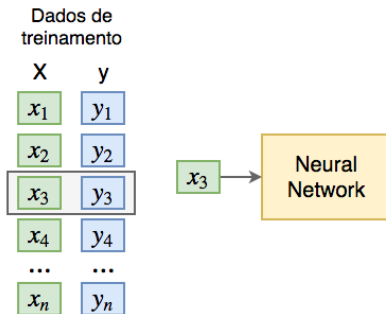
Treinando sem batch – Stochastic Gradient Descent



Batch normalization

O que é **batch**?

Treinando sem batch – Stochastic Gradient Descent



E assim por diante, para cada sample, por algumas épocas,...

Batch normalization

O que é **batch**?

Treinando com batch – Mini-batch Gradient Descent

Dados de
treinamento

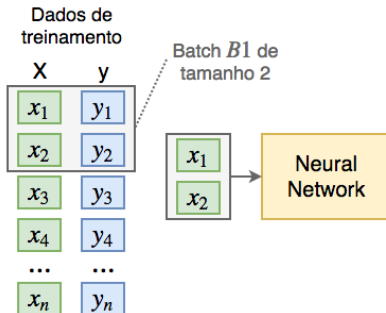
X	y
x_1	y_1
x_2	y_2
x_3	y_3
x_4	y_4
...	...
x_n	y_n

Neural
Network

Batch normalization

O que é **batch**?

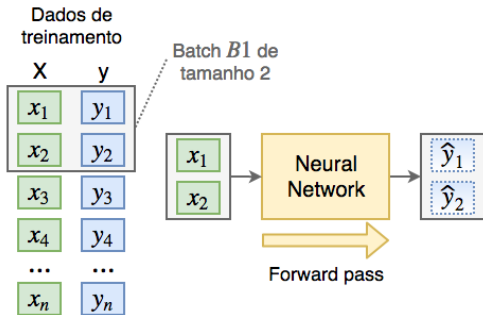
Treinando com batch – Mini-batch Gradient Descent



Batch normalization

O que é **batch**?

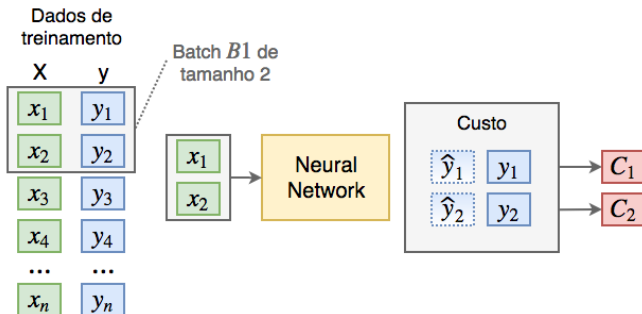
Treinando com batch – Mini-batch Gradient Descent



Batch normalization

O que é **batch**?

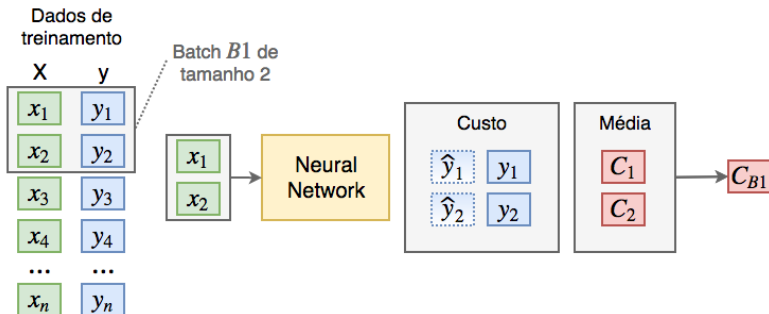
Treinando com batch – Mini-batch Gradient Descent



Batch normalization

O que é **batch**?

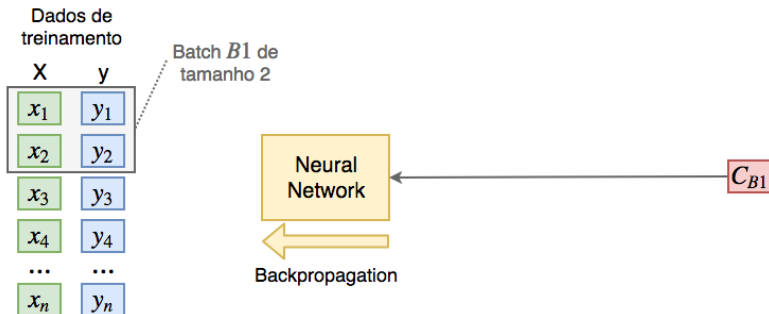
Treinando com batch – Mini-batch Gradient Descent



Batch normalization

O que é **batch**?

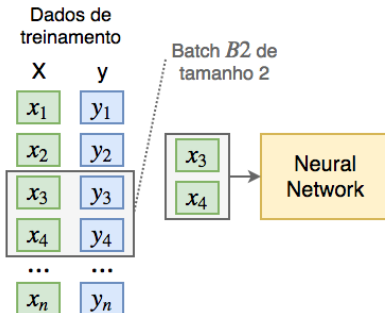
Treinando com batch – Mini-batch Gradient Descent



Batch normalization

O que é **batch**?

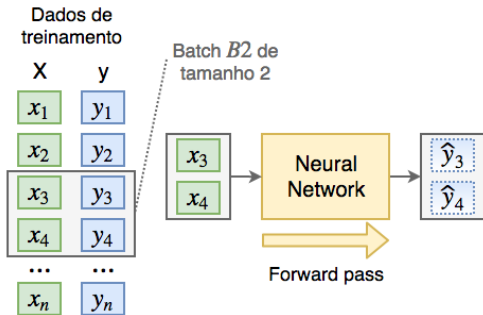
Treinando com batch – Mini-batch Gradient Descent



Batch normalization

O que é **batch**?

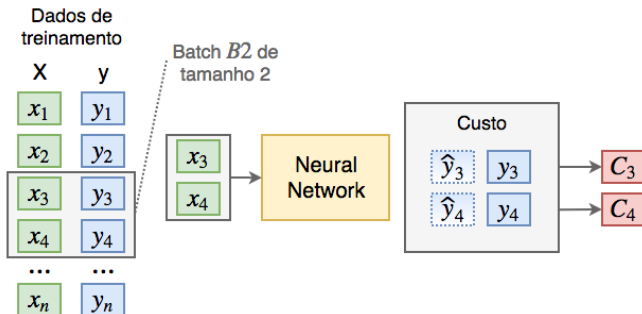
Treinando com batch – Mini-batch Gradient Descent



Batch normalization

O que é **batch**?

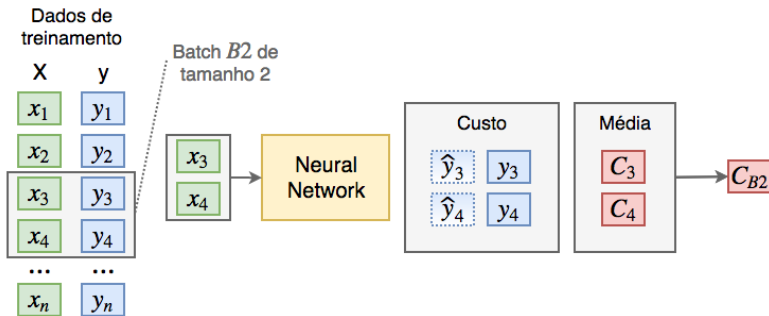
Treinando com batch – Mini-batch Gradient Descent



Batch normalization

O que é **batch**?

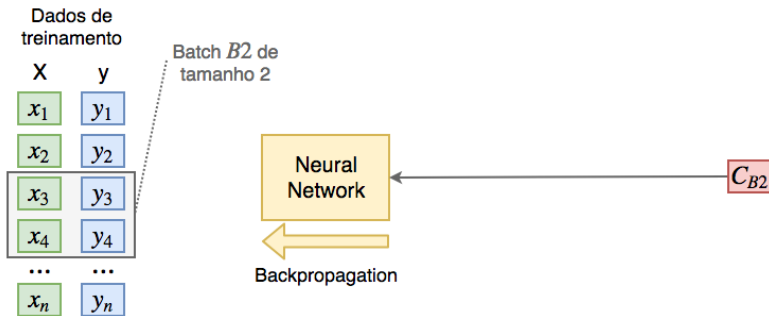
Treinando com batch – Mini-batch Gradient Descent



Batch normalization

O que é **batch**?

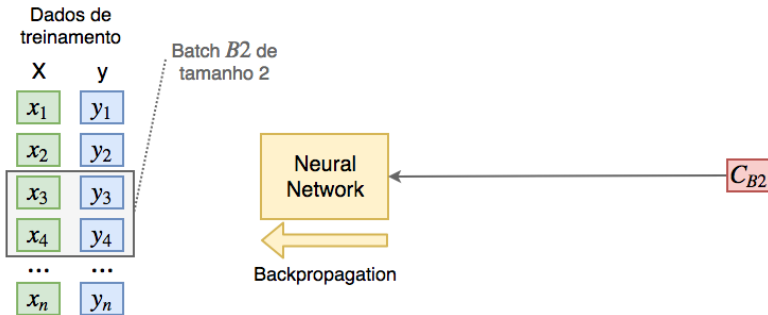
Treinando com batch – Mini-batch Gradient Descent



Batch normalization

O que é **batch**?

Treinando com batch – Mini-batch Gradient Descent



E assim por diante, para cada batch, por algumas épocas,...

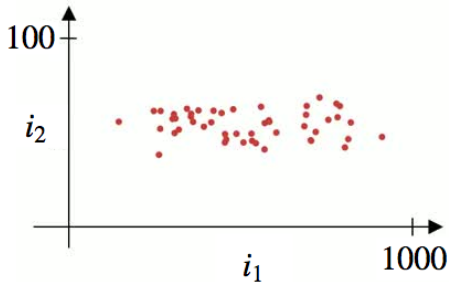
O que é **batch**?

Batch

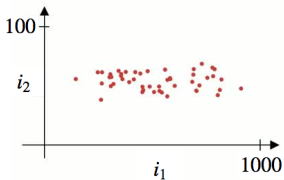
Quantidade de samples passados para a rede neural em uma única iteração.

O que é **normalização**?

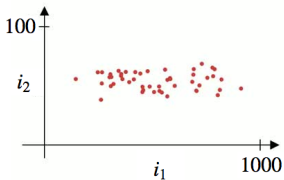
	i_1		i_2
	Idade		Saldo
x_1 :	[18	,	331]
x_2 :	[19	,	20]
x_3 :	[32	,	620]
x_4 :	[57	,	182]
	...		
x_n :	[15	,	99]



O que é **normalização**?

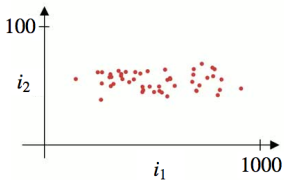


O que é **normalização**?



$$\mu = \frac{1}{n} \sum_{i=1}^n x_i$$

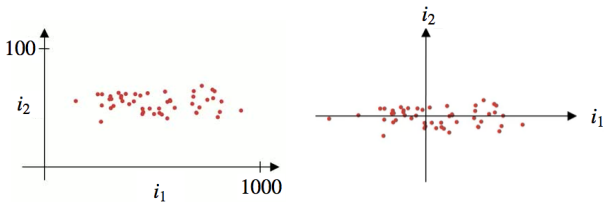
O que é **normalização**?



$$\mu = \frac{1}{n} \sum_{i=1}^n x_i$$

$$X = X - \mu$$

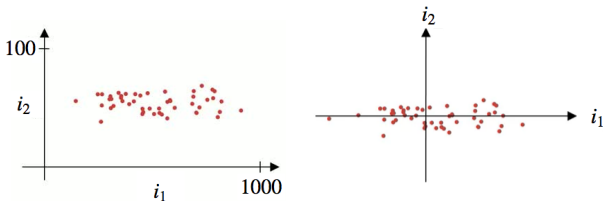
O que é **normalização**?



$$\mu = \frac{1}{n} \sum_{i=1}^n x_i$$

$$X = X - \mu$$

O que é **normalização**?

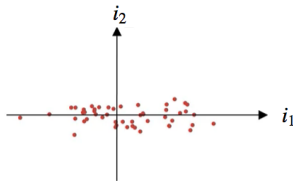
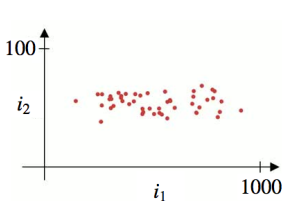


$$\mu = \frac{1}{n} \sum_{i=1}^n x_i$$

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^n x_i^2$$

$$X = X - \mu$$

O que é **normalização**?



$$\mu = \frac{1}{n} \sum_{i=1}^n x_i$$

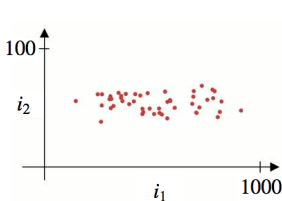
$$\sigma^2 = \frac{1}{n} \sum_{i=1}^n x_i^2$$

$$X = X - \mu$$

$$X = X / \sigma$$

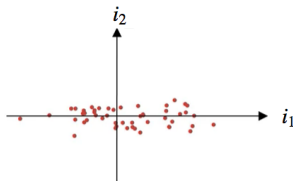
Batch normalization

O que é **normalização**?



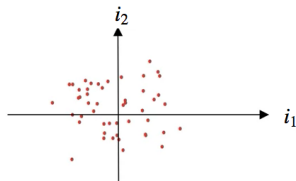
$$\mu = \frac{1}{n} \sum_{i=1}^n x_i$$

$$X = X - \mu$$



$$\sigma^2 = \frac{1}{n} \sum_{i=1}^n x_i^2$$

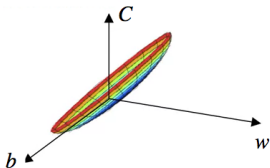
$$X = X / \sigma$$



O que é **normalização**?

Analisando a função de custo

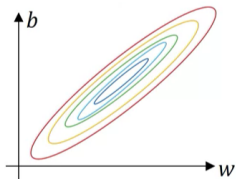
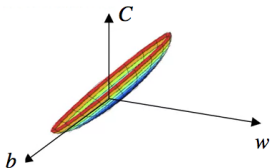
Sem normalização dos dados:



O que é **normalização**?

Analisando a função de custo

Sem normalização dos dados:

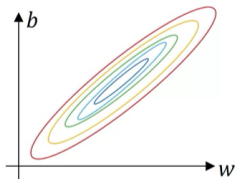
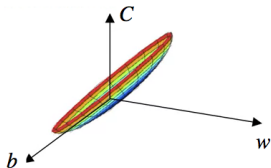


Batch normalization

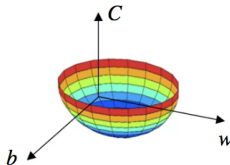
O que é **normalização**?

Analisando a função de custo

Sem normalização dos dados:



Com normalização dos dados:

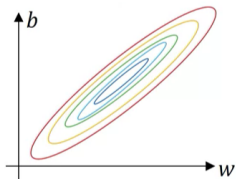
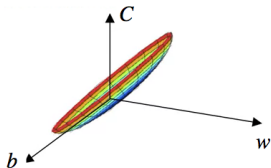


Batch normalization

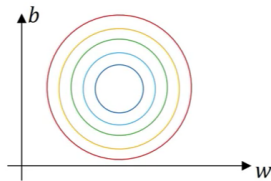
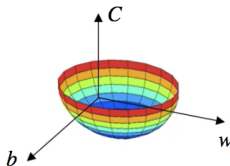
O que é **normalização**?

Analisando a função de custo

Sem normalização dos dados:



Com normalização dos dados:

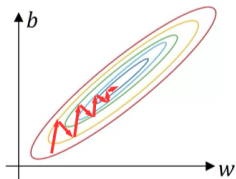
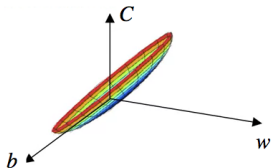


Batch normalization

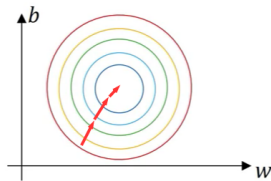
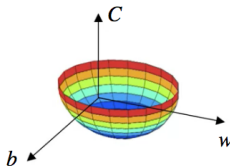
O que é **normalização**?

Analisando a função de custo

Sem normalização dos dados:



Com normalização dos dados:



Batch normalization

**INPUT
LAYER**



Input features

Batch normalization

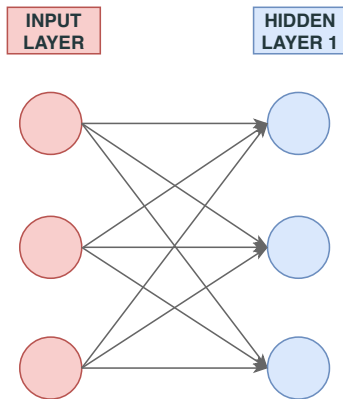
**INPUT
LAYER**



Input features

Data normalization!

Batch normalization

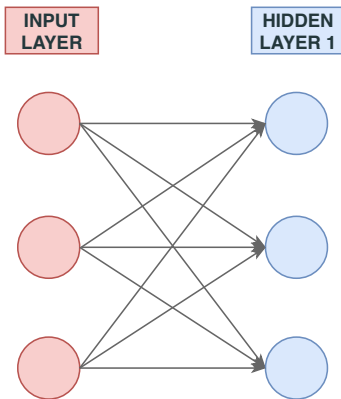


Input features

Novas features?

Data normalization!

Batch normalization



Input features

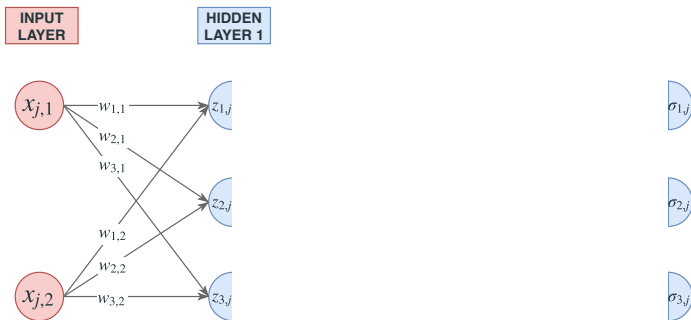
Novas features?

Data normalization!

Batch normalization!

Batch normalization

Durante o treinamento

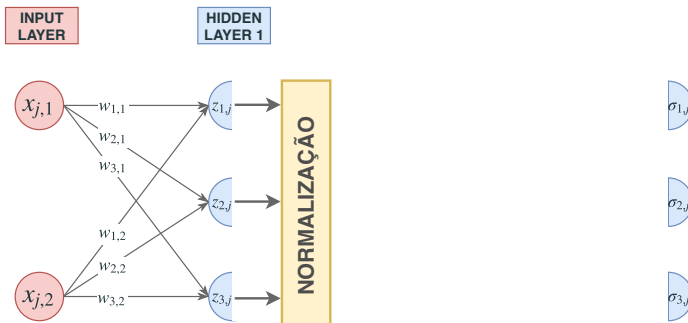


Para um batch $B = \{x_1, x_2, \dots, x_m\}$, o neurônio i produz $z_{i,j}$, para cada cada input $x_j \in B$:

$$z_{i,j} = w_i \cdot x_j + b_i.$$

Batch normalization

Durante o treinamento

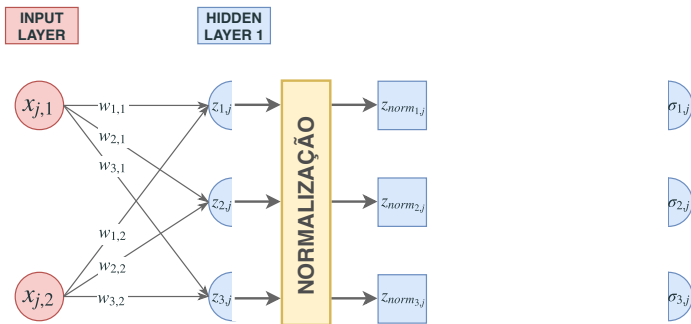


Calcular média e variância de $z_{i,j}$ para $j \in B$:

$$\mu_i = \frac{1}{m} \sum_{j=1}^m z_{i,j} \quad \sigma_i^2 = \frac{1}{m} \sum_{j=1}^m z_{i,j}^2 - \mu_i^2$$

Batch normalization

Durante o treinamento

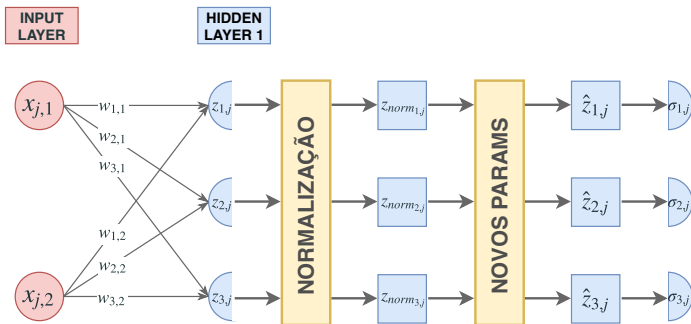


$$\text{Normalizar } z_{i,j}: z_{norm_{i,j}} = \frac{z_{i,j} - \mu_i}{\sqrt{\sigma_i^2 + \epsilon}}.$$

Bias $b_i = 0$, pois já seria cancelado pela subtração da média.

Batch normalization

Durante o treinamento



Novos parâmetros γ_i e β_i :

$$\hat{z}_{i,j} = \gamma_i z_{norm_{i,j}} + \beta_i$$

são aprendidos para não limitar a capacidade da rede.

Durante o teste

Utilizar médias e variâncias móveis calculadas durante o treinamento para cada neurônio.

Keras – Exemplo MNIST:

Modelo **sem** batch normalization:

```
# Creating model
model_without_bn = Sequential()

# Adjusting model structure
model_without_bn.add(Dense(256, activation="relu", input_shape=(784,)))
model_without_bn.add(Dense(128, activation="relu"))
model_without_bn.add(Dense(64, activation="relu"))
model_without_bn.add(Dense(10, activation="softmax"))
```

Batch normalization

Keras – Exemplo MNIST:

Modelo **sem** batch normalization:

```
# Creating model
model_without_bn = Sequential()

# Adjusting model structure
model_without_bn.add(Dense(256, activation="relu", input_shape=(784,)))
model_without_bn.add(Dense(128, activation="relu"))
model_without_bn.add(Dense(64, activation="relu"))
model_without_bn.add(Dense(10, activation="softmax"))
```

Modelo **com** batch normalization:

```
# Creating model
model_with_bn = Sequential()

# Adjusting model structure
model_with_bn.add(Dense(256, use_bias=False, input_shape=(784,)))
model_with_bn.add(BatchNormalization())
model_with_bn.add(Activation("relu"))

model_with_bn.add(Dense(128, use_bias=False))
model_with_bn.add(BatchNormalization())
model_with_bn.add(Activation("relu"))

model_with_bn.add(Dense(64, use_bias=False))
model_with_bn.add(BatchNormalization())
model_with_bn.add(Activation("relu"))

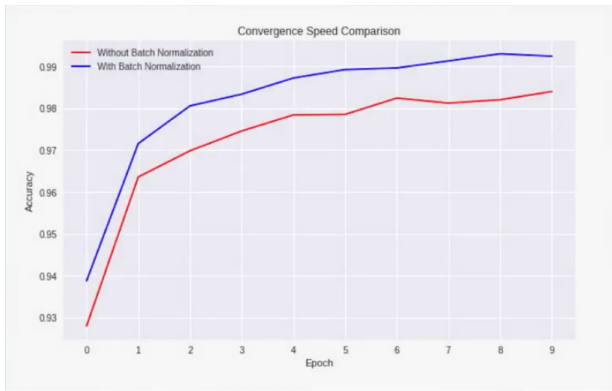
model_with_bn.add(Dense(10, activation="softmax"))
```

Keras – Exemplo MNIST:

- epochs = 10
- batch size = 128
- learning rate = 0.01
- data normalization: $X/255$
- weight init: `glorot_uniform`

Batch normalization

Keras – Exemplo MNIST:



Without batch norm test-acc: 0.9657

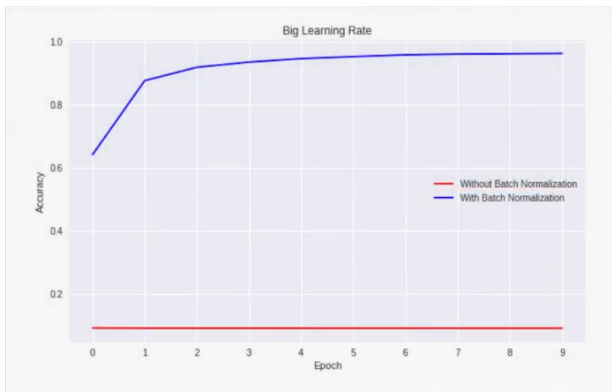
With batch norm test-acc: 0.9739

Keras – Exemplo MNIST:

- epochs = 10
- batch size = 128 \implies 1024
- learning rate = 0.01 \implies 1
- data normalization: $X/255$
- weight init: `glorot_uniform`

Batch normalization

Keras – Exemplo MNIST:



Without batch norm test-acc: 0.0892

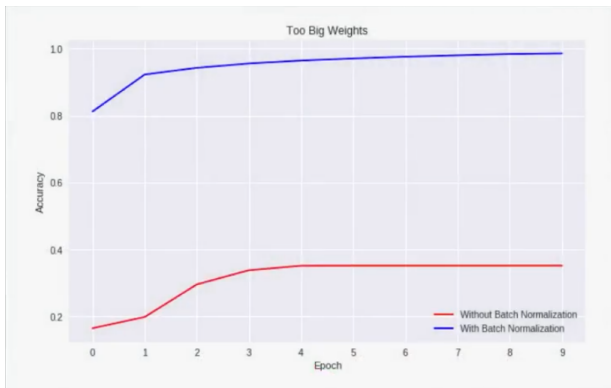
With batch norm test-acc: 0.9505

Keras – Exemplo MNIST:

- epochs = 10
- batch size = 128
- learning rate = 0.01
- data normalization: $X/255$
- weight init: `glorot_uniform` \implies
`RandomUniform(minval=-5, maxval=5)`

Batch normalization

Keras – Exemplo MNIST:



Without batch norm test-acc: 0.3525

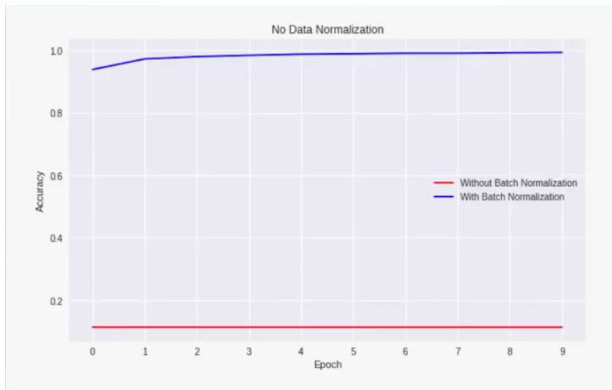
With batch norm test-acc: 0.9660

Keras – Exemplo MNIST:

- epochs = 10
- batch size = 128
- learning rate = 0.01
- data normalization: $X/255 \implies$ sem normalização
- weight init: `glorot_uniform`

Batch normalization

Keras – Exemplo MNIST:



Without batch norm test-acc: 0.1135

With batch norm test-acc: 0.9766

Vantagens da técnica de **batch normalization**:

- utilização de learning rates mais altos
- convergência do modelo em menos épocas
- inicialização de pesos de maneira não tão cuidadosa
- menor necessidade de normalização dos dados
- adição de regularização quando o batch não é tão grande

Transfer Learning

Sempre são necessários muitos dados para treinamento de uma CNN.



Sempre são necessários muitos dados para treinamento de uma CNN.



Transfer Learning

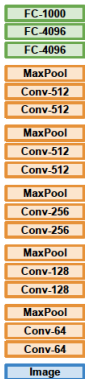
CNN treinada em dataset grande (p.ex., ImageNet)



Donahue et al, "DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition", ICML 2014
Razavian et al, "CNN Features Off-the-Shelf: An Astounding Baseline for Recognition", CVPR Workshops 2014

Transfer Learning

CNN treinada em dataset grande (p.ex., ImageNet)



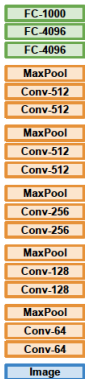
Transfer Learning com dataset pequeno e C classes



Donahue et al, "DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition", ICML 2014
Razavian et al, "CNN Features Off-the-Shelf: An Astounding Baseline for Recognition", CVPR Workshops 2014

Transfer Learning

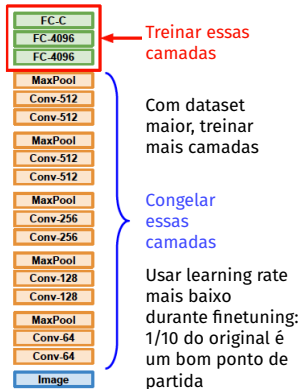
CNN treinada em dataset grande (p.ex., ImageNet)



Transfer Learning com dataset pequeno e C classes



... ou com dataset maior



Transfer Learning



mais específico

mais genérico

	dataset bem similar	dataset bem diferente
poucos dados	?	?
muitos dados	?	?

Transfer Learning



mais específico

mais genérico


	dataset bem similar	dataset bem diferente
poucos dados	usar classificador linear como última camada	?
muitos dados	Finetuning de algumas camadas	?

Transfer Learning



mais específico

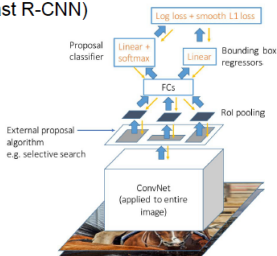
mais genérico

	dataset bem similar	dataset bem diferente
poucos dados	usar classificador linear como última camada	 Tentar classificador linear em diferentes estágios
muitos dados	Finetuning de algumas camadas	Finetuning de muitas camadas

Transfer Learning

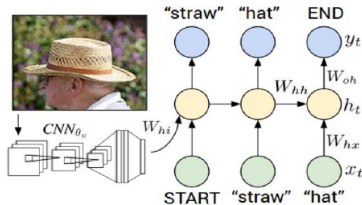
Transfer learning é o mais comum na prática.

Object Detection (Fast R-CNN)



Girshick, "Fast R-CNN". ICCV 2015
Figure copyright Ross Girshick, 2015. Reproduced with permission.

Image Captioning: CNN + RNN

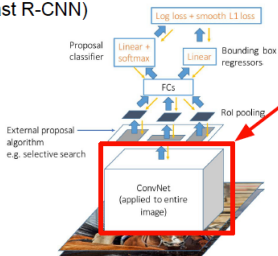


Karpathy and Fei-Fei, "Deep Visual-Semantic Alignments for Generating Image Descriptions". CVPR 2015
Figure copyright IEEE, 2015. Reproduced for educational purposes.

Transfer Learning

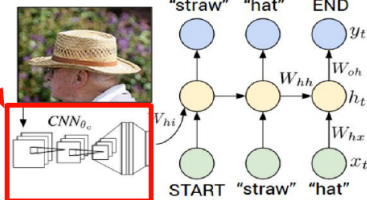
Transfer learning é o mais comum na prática.

Object Detection (Fast R-CNN)



**CNN pré-treinada
na Imagenet**

Image Captioning: CNN + RNN



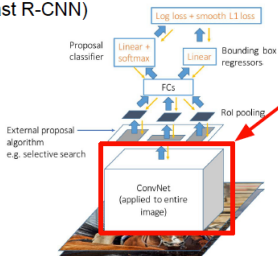
Girshick, "Fast R-CNN". ICCV 2015
Figure copyright Ross Girshick, 2015. Reproduced with permission.

Karpathy and Fei-Fei, "Deep Visual-Semantic Alignments for
Generating Image Descriptions". CVPR 2015
Figure copyright IEEE, 2015. Reproduced for educational purposes.

Transfer Learning

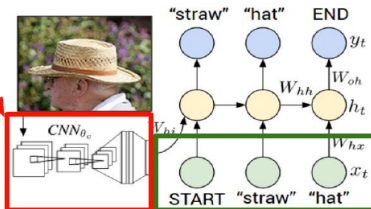
Transfer learning é o mais comum na prática.

Object Detection (Fast R-CNN)



**CNN pré-treinada
na Imagenet**

Image Captioning: CNN + RNN



**Vetor de palavras
pré-treinado na
word2vec**

Karpathy and Fei-Fei, "Deep Visual-Semantic Alignments for Generating Image Descriptions", CVPR 2015
Figure copyright IEEE, 2015. Reproduced for educational purposes.

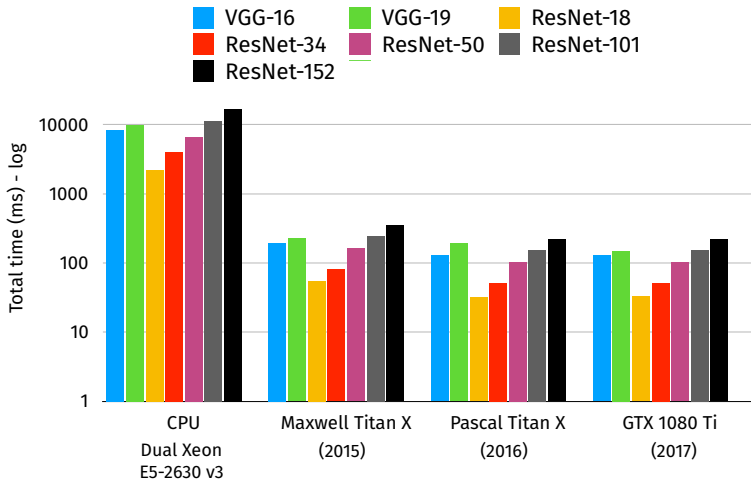
Girshick, "Fast R-CNN", ICCV 2015
Figure copyright Ross Girshick, 2015. Reproduced with permission.

Tem um tarefa de interesse, mas o dataset possui $< \sim 1M$ imagens?

- Encontre um dataset grande para uma tarefa similar
- Treine uma CNN com este dataset grande
- Faça transfer learning para a sua tarefa

CPU vs. GPU

CPU vs. GPU



Fonte: <https://github.com/jcjohnson/cnn-benchmarks>

O uso de GPUs é altamente recomendado!

O treinamento em GPU é de 49× a 74× mais rápido do que em CPU.