NORMALIZAÇÃO EM LOTE

Zenilton K. G. Patrocínio Jr
zenilton@pucminas.br

Normalização em Lote (*Batch Normalization*)

“Você quer ativações gaussianas unitárias? Apenas faça-as assim.

[Ioffe and Szegedy, 2015]

Normalização em Lote (*Batch Normalization*)

“Você quer ativações gaussianas unitárias? Apenas faça-as assim.

[Ioffe and Szegedy, 2015]

Considere um lote de ativações em uma camada qualquer.

Para fazer com que cada dimensão se comporte como gaussiana unitária, aplica-se:

$$\widehat{x}^{(k)} = \frac{x^{(k)} - \mathrm{E}[x^{(k)}]}{\sqrt{\mathrm{Var}[x^{(k)}]}}$$

Normalização em Lote (*Batch Normalization*)

“Você quer ativações gaussianas unitárias? Apenas faça-as assim.

[Ioffe and Szegedy, 2015]

Considere um lote de ativações em uma camada qualquer.

Para fazer com que cada dimensão se comporte como gaussiana unitária, aplica-se:

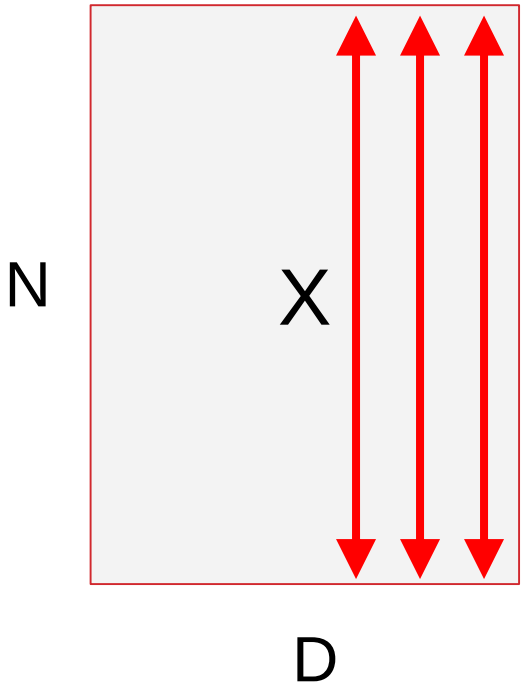
$$\widehat{x}^{(k)} = \frac{x^{(k)} - \mathrm{E}[x^{(k)}]}{\sqrt{\mathrm{Var}[x^{(k)}]}}$$

→ Esta é uma função diferenciável comum ...

Normalização em Lote (*Batch Normalization*)

"Você quer ativações gaussianas unitárias? Apenas faça-as assim.

[Ioffe and Szegedy, 2015]

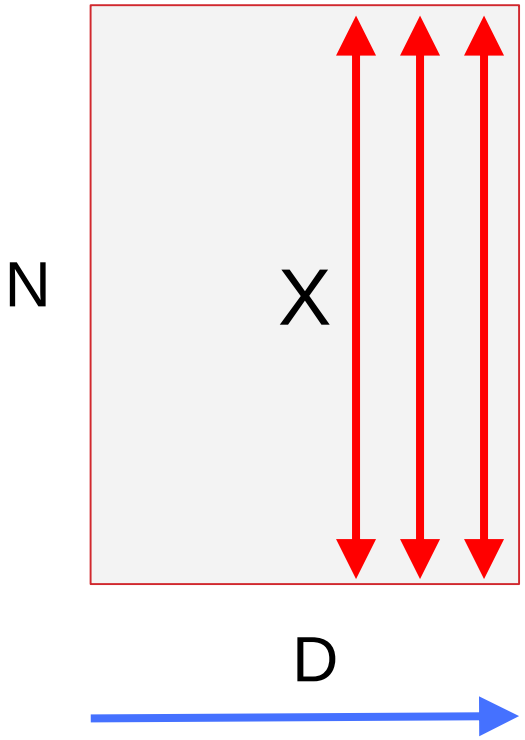


1. Calcular a média e a variância empírica para cada dimensão de forma independente

Normalização em Lote (*Batch Normalization*)

"Você quer ativações gaussianas unitárias? Apenas faça-as assim.

[Ioffe and Szegedy, 2015]

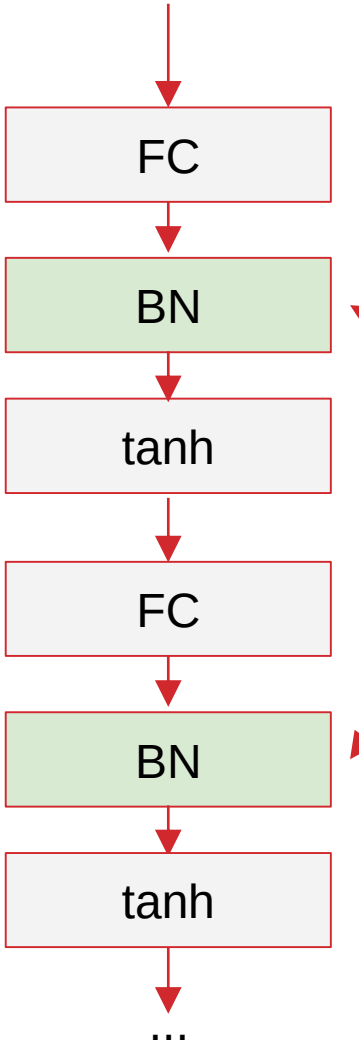


1. Calcular a média e a variância empírica para cada dimensão de forma independente

2. Normalizar

$$\hat{x}^{(k)} = \frac{x^{(k)} - \mathrm{E}[x^{(k)}]}{\sqrt{\mathrm{Var}[x^{(k)}]}}$$

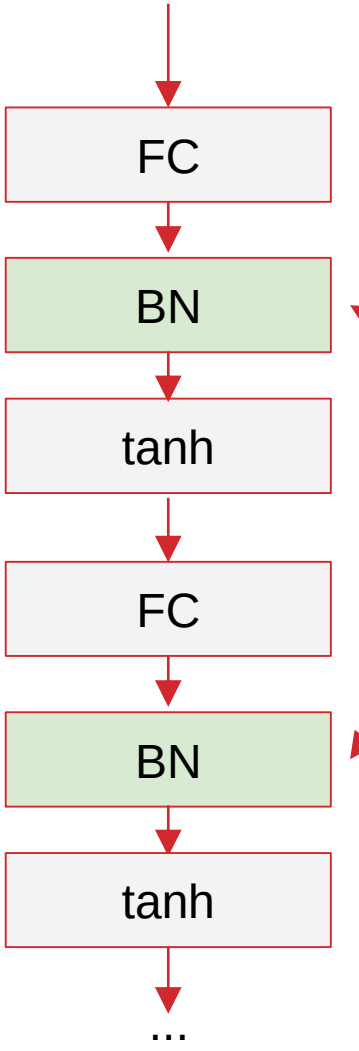
Normalização em Lote (*Batch Normalization*)



Normalmente inserida após as camadas completamente conectadas (ou camadas convolucionais), mas antes da não linearidade.

$$\widehat{x}^{(k)} = \frac{x^{(k)} - \mathrm{E}[x^{(k)}]}{\sqrt{\mathrm{Var}[x^{(k)}]}}$$

Normalização em Lote (*Batch Normalization*)



Normalmente inserida após as camadas completamente conectadas (ou camadas convolucionais), mas antes da não linearidade.

Problema: deseja-se realmente uma entrada gaussiana unitária para a não-linearidade?

$$\hat{x}^{(k)} = \frac{x^{(k)} - \mathrm{E}[x^{(k)}]}{\sqrt{\mathrm{Var}[x^{(k)}]}}$$

Normalização em Lote (*Batch Normalization*)

Solução!

Normalizar:

$$\widehat{x}^{(k)} = \frac{x^{(k)} - \mathrm{E}[x^{(k)}]}{\sqrt{\mathrm{Var}[x^{(k)}]}}$$

Normalização em Lote (*Batch Normalization*)

Solução!

Normalizar:

$$\widehat{x}^{(k)} = \frac{x^{(k)} - \mathrm{E}[x^{(k)}]}{\sqrt{\mathrm{Var}[x^{(k)}]}}$$

E, em seguida, permitir que a rede ajuste a saída para outro intervalo, se quiser:

$$y^{(k)} = \gamma^{(k)}\widehat{x}^{(k)} + \beta^{(k)}$$

em que $\gamma^{(k)}$ e $\beta^{(k)}$ devem ser aprendidos pela rede

Normalização em Lote (*Batch Normalization*)

Solução!

Normalizar:

$$\widehat{x}^{(k)} = \frac{x^{(k)} - \mathrm{E}[x^{(k)}]}{\sqrt{\mathrm{Var}[x^{(k)}]}}$$

E, em seguida, permitir que a rede ajuste a saída para outro intervalo, se quiser:

$$y^{(k)} = \gamma^{(k)}\widehat{x}^{(k)} + \beta^{(k)}$$

em que $\gamma^{(k)}$ e $\beta^{(k)}$ devem ser aprendidos pela rede

Observe que a rede poderia aprender, de modo que:

$$\gamma^{(k)} = \sqrt{\mathrm{Var}[x^{(k)}]}$$

$$\beta^{(k)} = \mathrm{E}[x^{(k)}]$$

e, assim, recuperar o mapeamento de identidade

Normalização em Lote (*Batch Normalization*)

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β

Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma^2_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma^2_{\mathcal{B}} + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma\widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

- Melhora o fluxo gradiente através da rede

Normalização em Lote (*Batch Normalization*)

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β

Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma^2_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma^2_{\mathcal{B}} + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma\widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

- Melhora o fluxo gradiente através da rede
- Permite taxas de aprendizagem mais altas

Normalização em Lote (*Batch Normalization*)

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β

Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma^2_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma^2_{\mathcal{B}} + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma\widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

- Melhora o fluxo gradiente através da rede
- Permite taxas de aprendizagem mais altas
- Reduz a forte dependência da inicialização

Normalização em Lote (*Batch Normalization*)

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β

Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma^2_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma^2_{\mathcal{B}} + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma\widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

- Melhora o fluxo gradiente através da rede
- Permite taxas de aprendizagem mais altas
- Reduz a forte dependência da inicialização
- Atua como uma forma de regularização diferente, reduzindo talvez a necessidade de *dropout*

Normalização em Lote (*Batch Normalization*)

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β

Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma^2_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma^2_{\mathcal{B}} + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma\widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

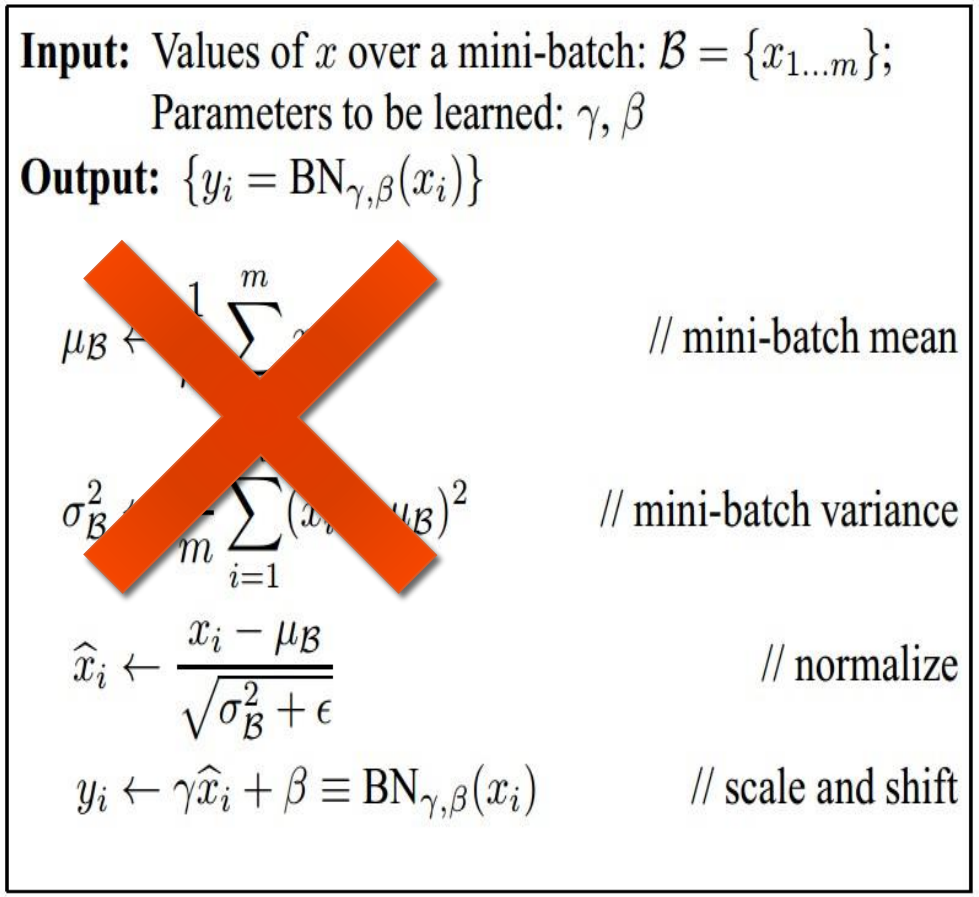
- Melhora o fluxo gradiente através da rede
- Permite taxas de aprendizagem mais altas
- Reduz a forte dependência da inicialização
- Atua como uma forma de regularização diferente, reduzindo talvez a necessidade de *dropout*

Desnormalização!!

Aprende-se γ e β (mesmas dims que μ e σ^2)

Permite (se necessário) aprender o mapeamento identidade!

Normalização em Lote (*Batch Normalization*)



Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β

Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m}$ // mini-batch mean

$\sigma^2_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}(x_i - \mu_{\mathcal{B}})^2$ // mini-batch variance

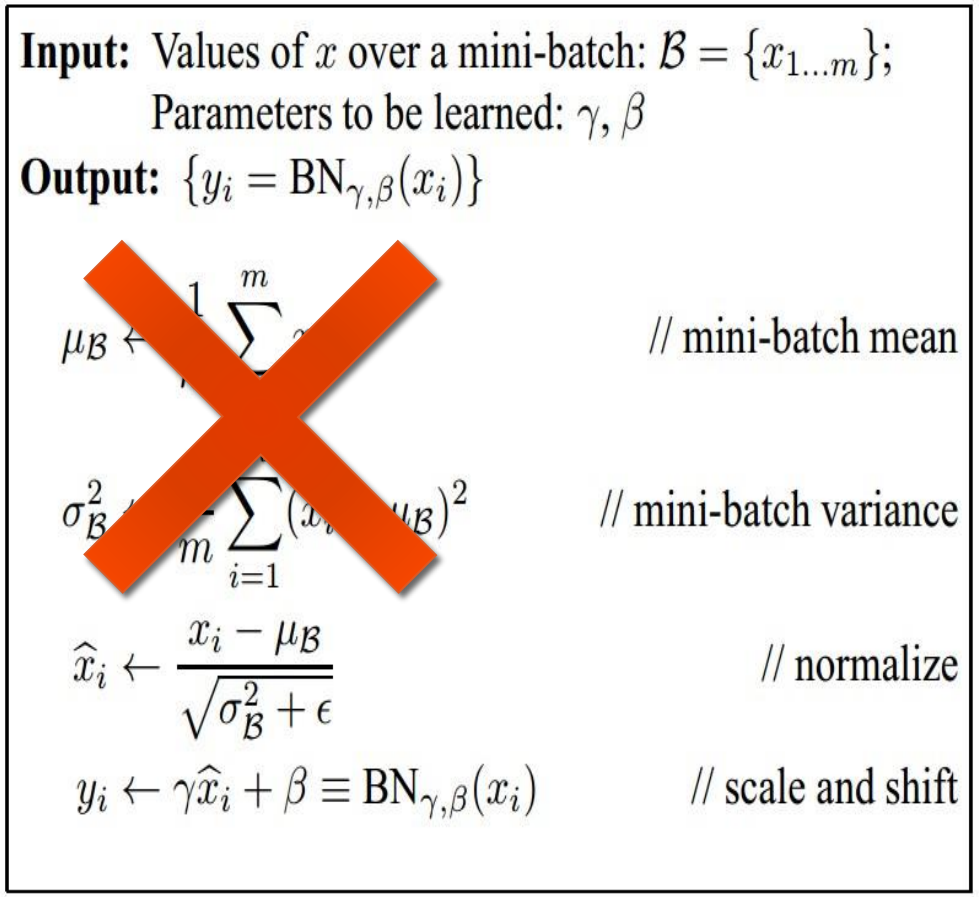
$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma^2_{\mathcal{B}} + \epsilon}}$ // normalize

$y_i \leftarrow \gamma\widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i)$ // scale and shift

Nota: no momento do **teste**, a camada de normalização em lote funciona diferente:

- A **média e a variância não são calculadas** com base no lote

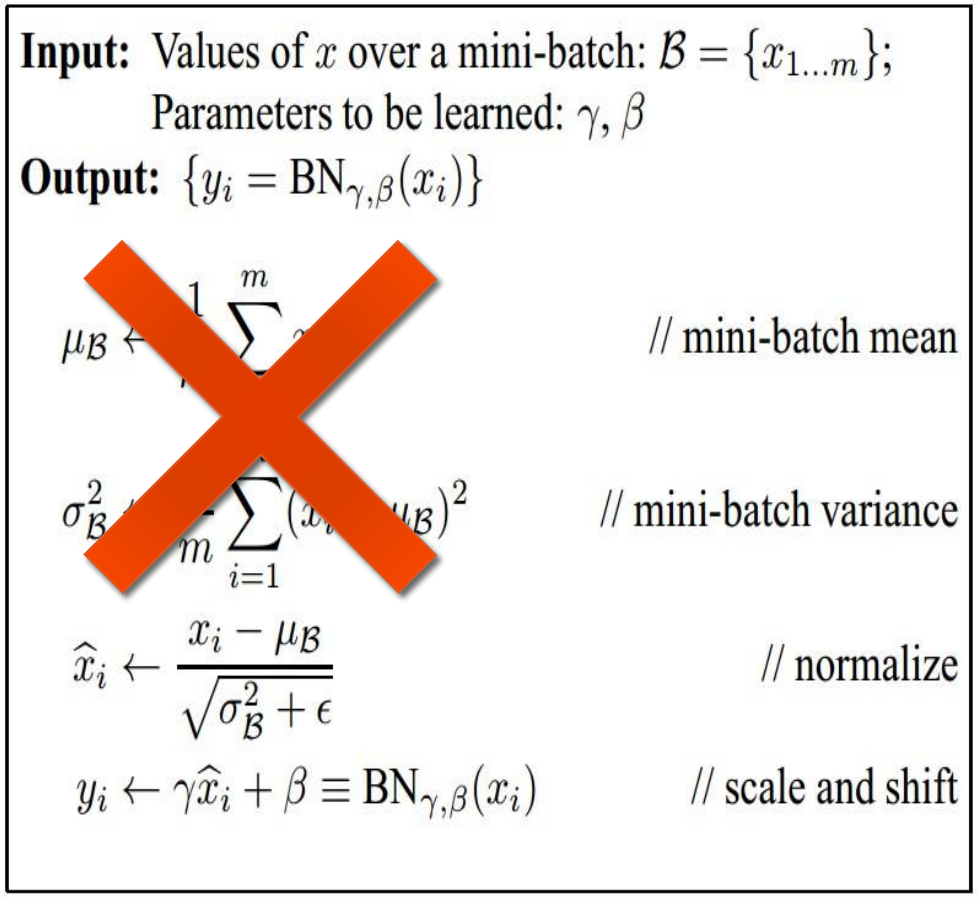
Normalização em Lote (*Batch Normalization*)



Nota: no momento do **teste**, a camada de normalização em lote funciona diferente**:**

- A **média e a variância não são calculadas** com base no lote
- Em vez disso, **utiliza-se um único par fixo de média e variância** empírica de ativações obtido durante o treinamento

Normalização em Lote (*Batch Normalization*)



Nota: no momento do **teste**, a camada de normalização em lote funciona diferente**:**

- A **média e a variância não são calculadas** com base no lote
- Em vez disso, **utiliza-se um único par fixo de média e variância** empírica de ativações obtido durante o treinamento
- Por exemplo, pode-se estimar o par durante o treinamento por meio de médias móveis