RNP-VC

Aula 4: Os 300 hiperparâmetros

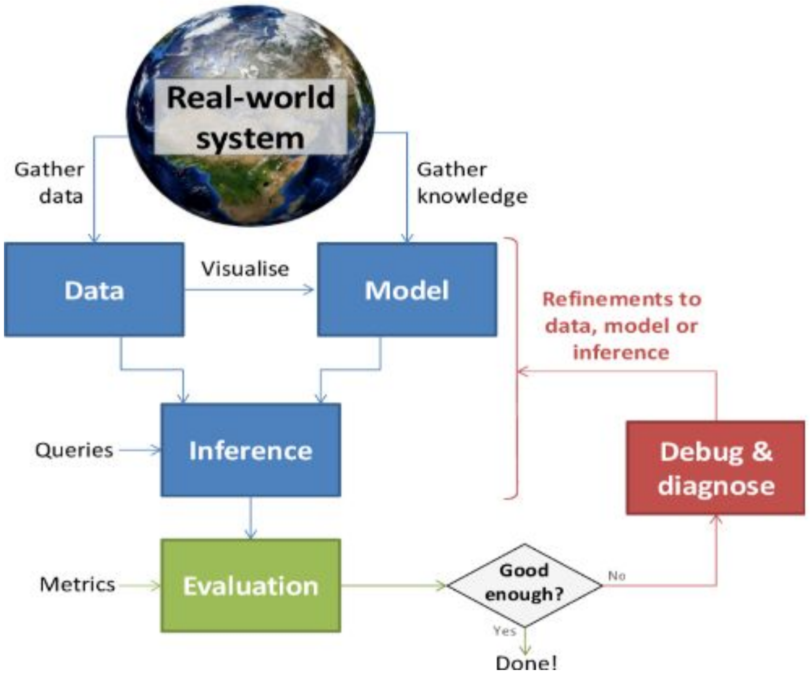
Lucas Pereira, Rafael Teixeira, Lucas Assis, Anderson Soares
Instituto de Informática
Universidade Federal de Goiás (UFG)

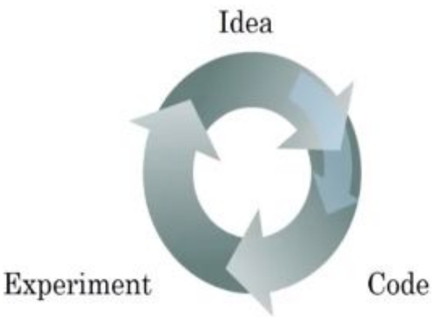
Sumário

- No último episódio...
- Hiperparâmetros
- No próximo episódio...

No último episódio...

- O ciclo tentador...



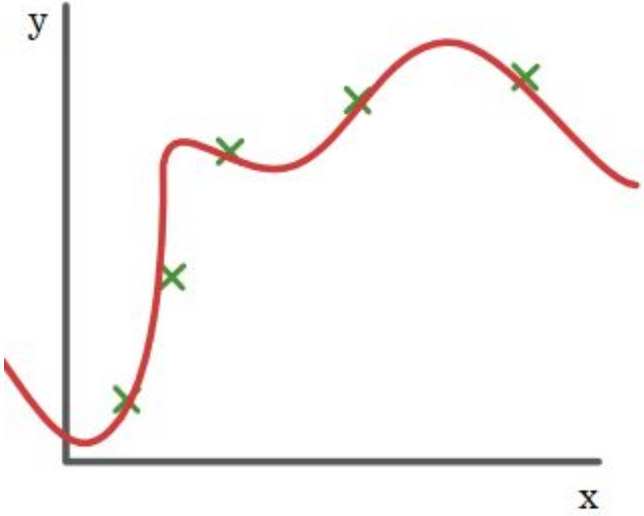


No último episódio...

- Divisão treino/validação/teste

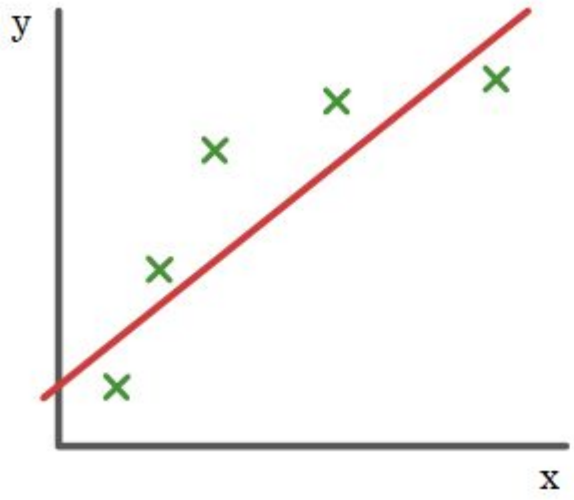
No último episódio...

- Threshold muito pequeno: overfitting



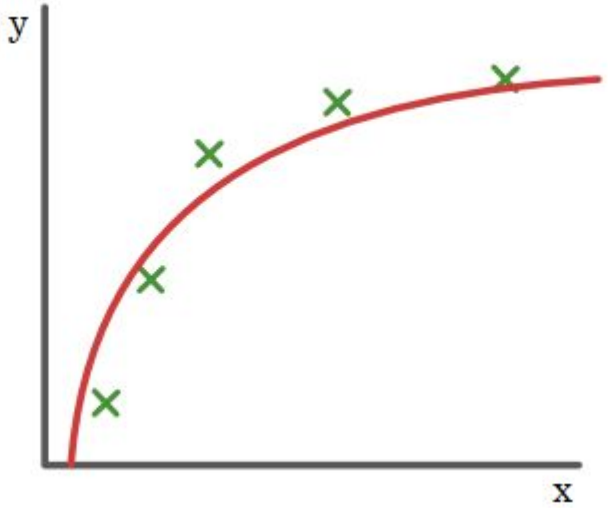
No último episódio...

- Threshold muito grande: underfitting



No último episódio...

- Threshold ideal



Hiperparâmetros

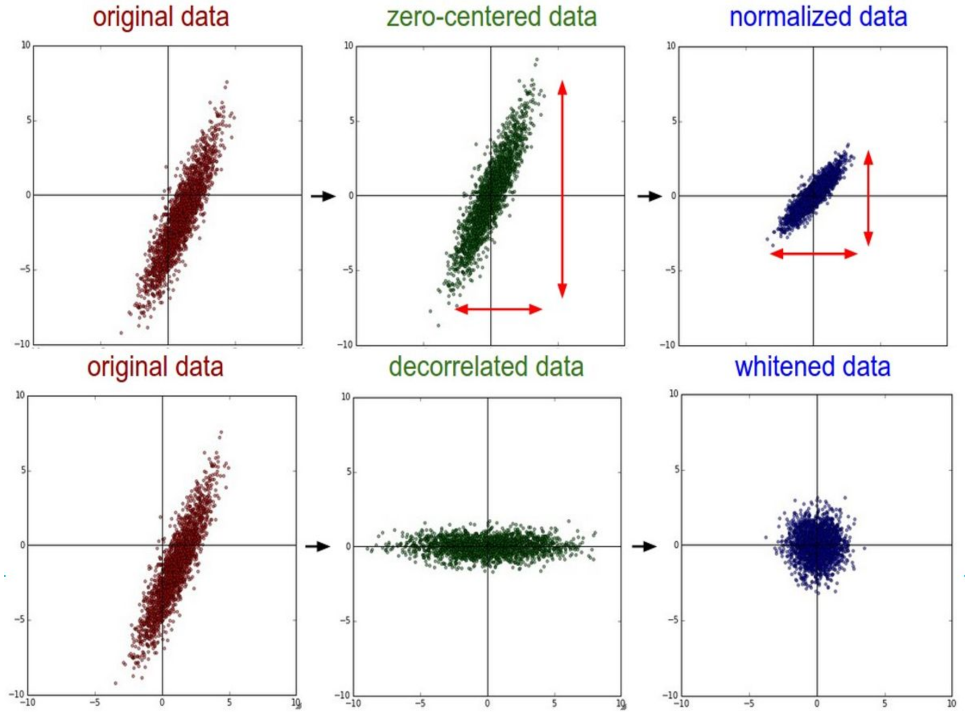
- Parâmetros:

$$W^{[1]}, b^{[1]}, W^{[2]}, b^{[2]}, W^{[3]}, b^{[3]} \ ...$$

- Hiperparâmetros:
    - Pré-processamento
    - Arquitetura
    - Função de ativação
    - Inicialização de parâmetros
    - Algoritmo de otimização
    - Taxa de aprendizado (learning rate)
    - Técnicas de regularização
    - [...]

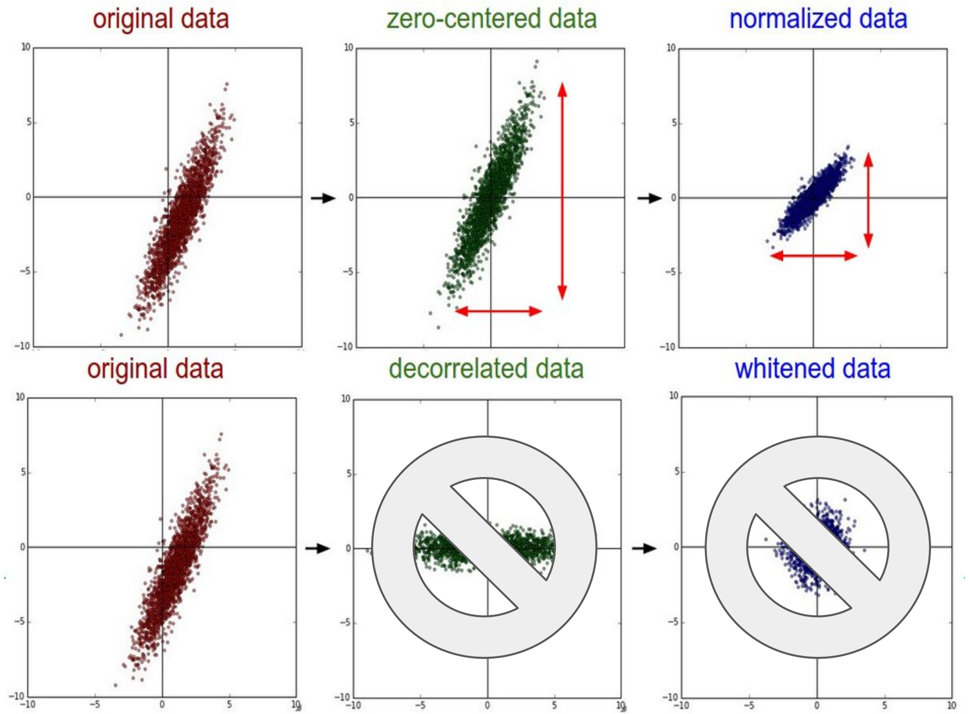
Hiperparâmetros

- Pré-processamento



Hiperparâmetros

- Pré-processamento (para imagens)




```
X -= np.mean(X, axis = 0)
```

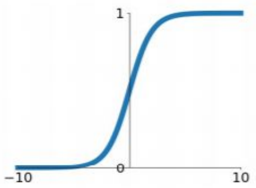
```
X /= np.std(X, axis = 0)
```

Hiperparâmetros

- Função de ativação

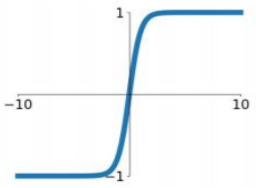
Sigmoid

$\sigma(x) = \frac{1}{1+e^{-x}}$



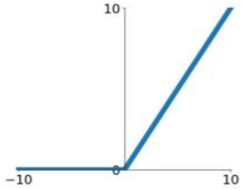
tanh

$\tanh(x)$



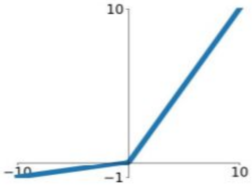
ReLU

$\max(0, x)$



Leaky ReLU

$\max(0.1x, x)$

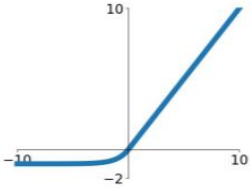


Maxout

$\max(w_1^T x + b_1, w_2^T x + b_2)$

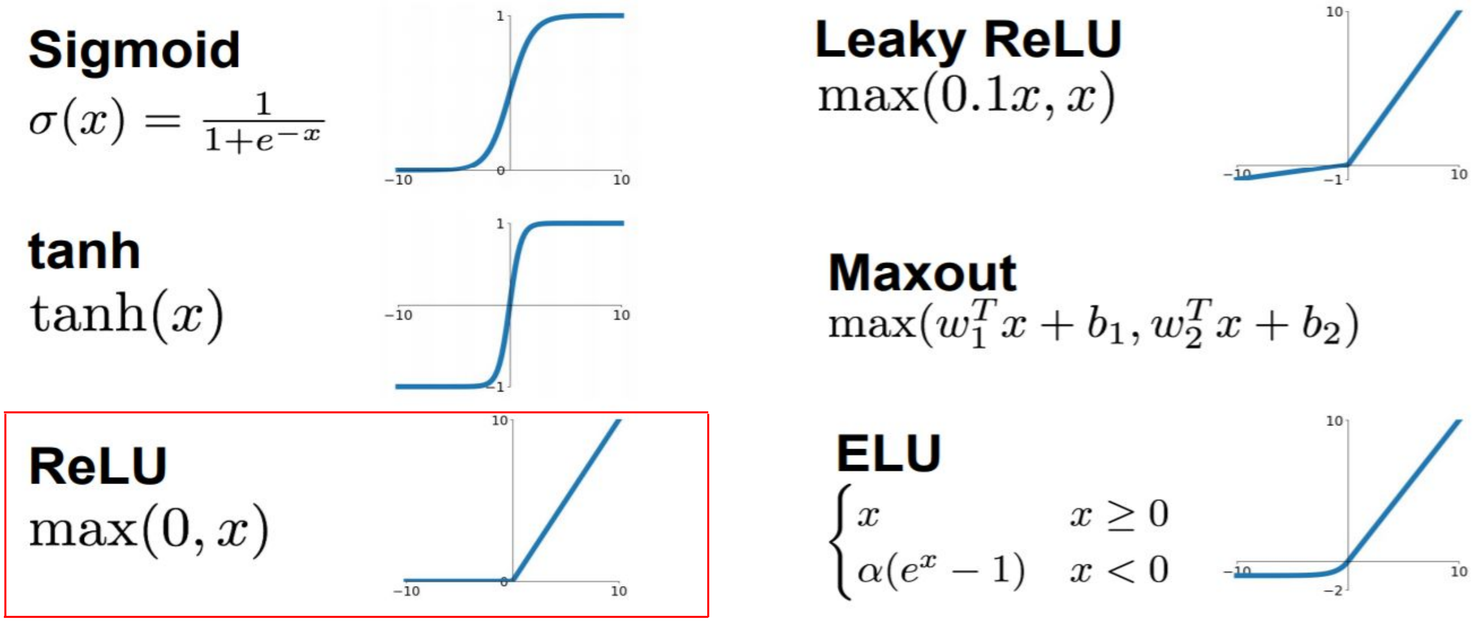
ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



Hiperparâmetros

- Função de ativação



Hiperparâmetros

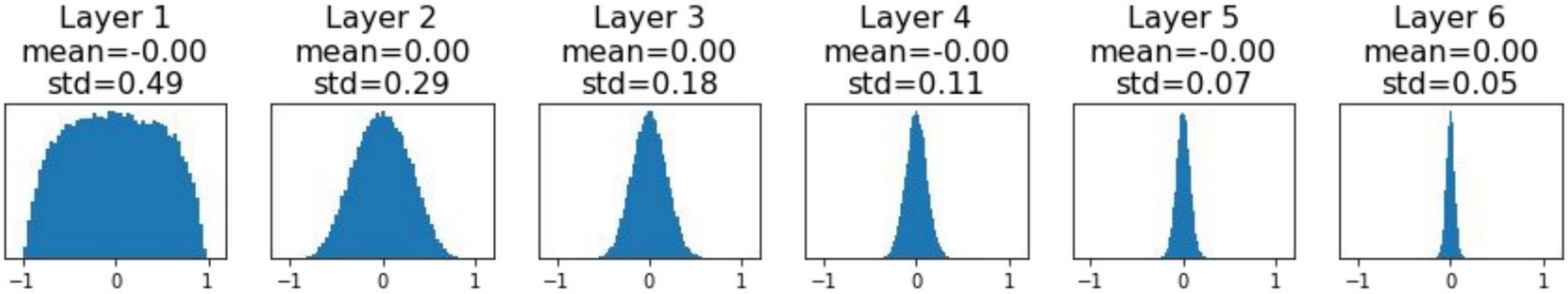
- Inicialização de parâmetros
  - Iniciar tudo com 0
  - Iniciar tudo com o mesmo valor > 0
  - Iniciar tudo com o mesmo valor < 0
  - Iniciar com valores aleatórios com alta variância
  - Iniciar com valores aleatórios com baixa variância

Hiperparâmetros

- Inicialização de parâmetros
    - Iniciar tudo com 0
    - Iniciar tudo com o mesmo valor > 0
    - Iniciar tudo com o mesmo valor < 0
    - Iniciar com valores aleatórios com alta variância
    - Iniciar com valores aleatórios com baixa variância

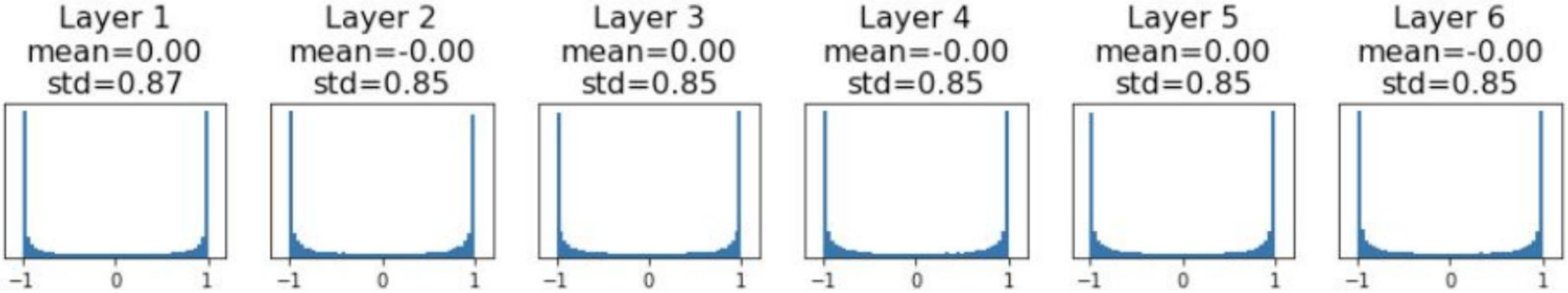
Hiperparâmetros

- Inicialização de parâmetros
  - W = 0.01*np.random.randn(D,H) (função de ativação: tanh)



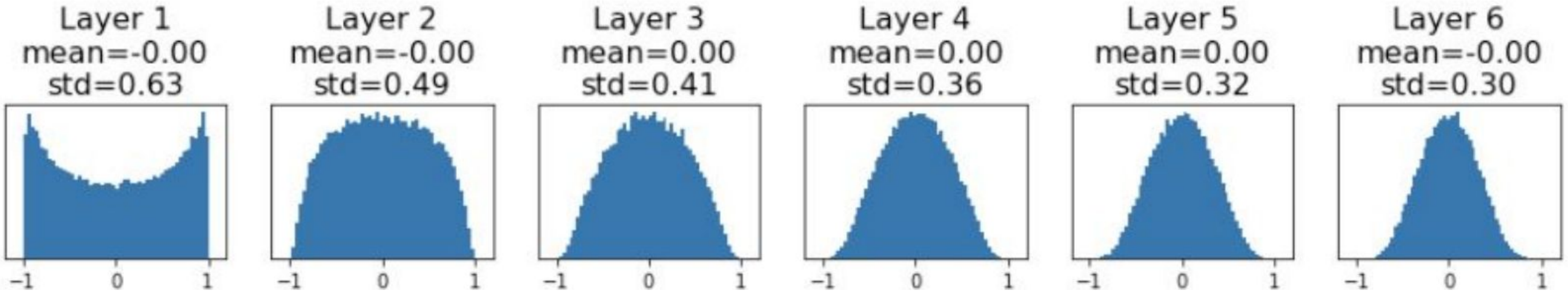
Hiperparâmetros

- Inicialização de parâmetros
  - W = 0.05*np.random.randn(D,H) (função de ativação: tanh)



Hiperparâmetros

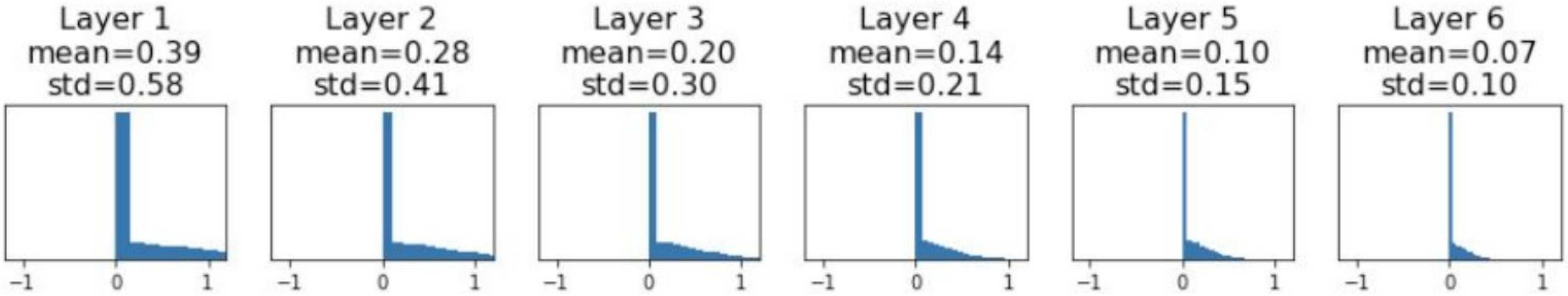
- Inicialização de parâmetros
    - W = (1./sqrt(D))*np.random.randn(D,H) (função de ativação: tanh) (Inicialização Xavier)



Glorot and Bengio, “Understanding the difficulty of training deep feedforward neural networks”, AISTAT 2010

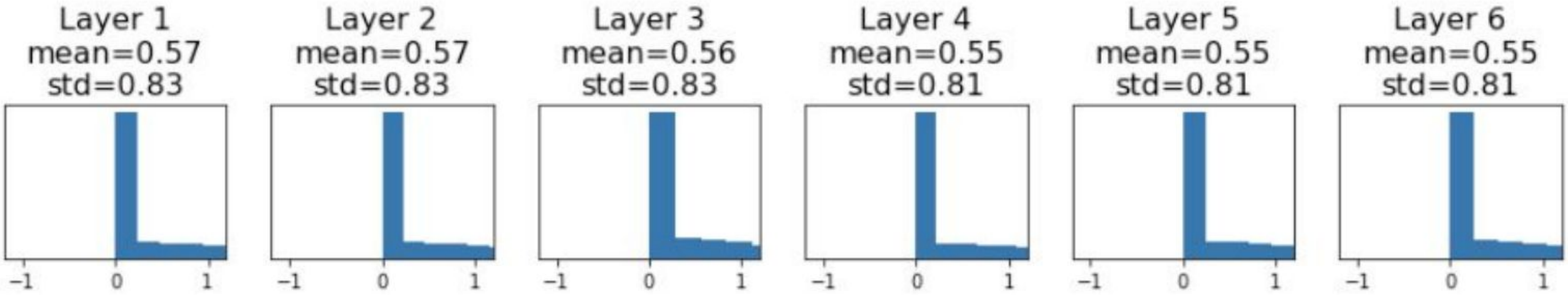
Hiperparâmetros

- Inicialização de parâmetros
    - W = (1./sqrt(D))*np.random.randn(D,H) (função de ativação: ReLU) (Inicialização Xavier)



Hiperparâmetros

- Inicialização de parâmetros
  - W = (1./sqrt(D/2))*np.random.randn(D,H) (função de ativação: ReLU) (Inicialização He)



He et al, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification", ICCV 2015

Hiperparâmetros

- Inicialização de parâmetros
  - Campo de pesquisa ativo

Understanding the difficulty of training deep feedforward neural networks
by Glorot and Bengio, 2010

Exact solutions to the nonlinear dynamics of learning in deep linear neural networks by Saxe et al, 2013

Random walk initialization for training very deep feedforward networks by Sussillo and Abbott, 2014

Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification by He et al., 2015

Data-dependent Initializations of Convolutional Neural Networks by Krähenbühl et al., 2015

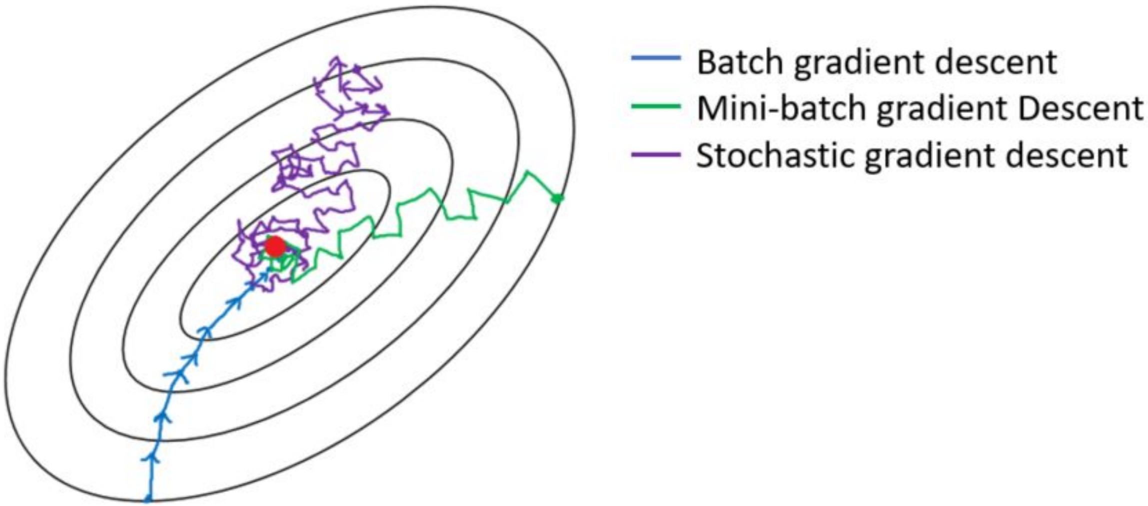
All you need is a good init, Mishkin and Matas, 2015

Fixup Initialization: Residual Learning Without Normalization, Zhang et al, 2019

The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks, Frankle and Carbin, 2019

Hiperparâmetros

- Algoritmos de otimização
  - Stochastic Gradient Descent (SGD)



Hiperparâmetros

- Algoritmos de otimização
  - (mini batch) Stochastic Gradient Descent (SGD)
    - batch_size < len(dataset)
  - SGD com Momento

Hiperparâmetros

- Algoritmos de otimização
    - (mini batch) Stochastic Gradient Descent (SGD)
        - batch_size < len(dataset)
    - SGD com Momento

```
v = mu * v - learning_rate * dx # integrate velocity
x += v # integrate position
```

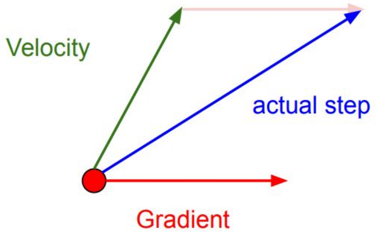
Hiperparâmetros

- Algoritmos de otimização
  - (mini batch) Stochastic Gradient Descent (SGD)
    - batch_size < len(dataset)
  - SGD com Momento

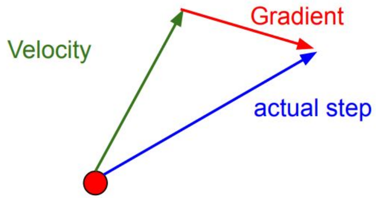
```
v = mu * v - learning_rate * dx # integrate velocity
x += v # integrate position
```

  - SGD com Momento de Nesterov

Momentum update:



Nesterov Momentum



Hiperparâmetros

- Algoritmos de otimização
    - (mini batch) Stochastic Gradient Descent (SGD)
        - batch_size < len(dataset)
    - SGD com Momento
    - SGD com Momento de Nesterov
    - Adagrad

```
cache += dx**2
x += - learning_rate * dx / (np.sqrt(cache) + eps)
```

Hiperparâmetros

- Algoritmos de otimização
  - (mini batch) Stochastic Gradient Descent (SGD)
    - batch_size < len(dataset)
  - SGD com Momento
  - SGD com Momento de Nesterov
  - Adagrad

```
cache += dx**2
x += - learning_rate * dx / (np.sqrt(cache) + eps)
```

  - RMSProp

```
cache = decay_rate * cache + (1 - decay_rate) * dx**2
x += - learning_rate * dx / (np.sqrt(cache) + eps)
```

Hiperparâmetros

- Algoritmos de otimização
  - (mini batch) Stochastic Gradient Descent (SGD)
    - batch_size < len(dataset)
  - SGD com Momento
  - SGD com Momento de Nesterov
  - Adagrad
  - RMSProp
  - Adam (ada + m)

Hiperparâmetros

- Algoritmos de otimização
    - (mini batch) Stochastic Gradient Descent (SGD)
        - batch_size < len(dataset)
    - SGD com Momento
    - SGD com Momento de Nesterov
    - Adagrad
    - RMSProp
    - Adam (ada + m)

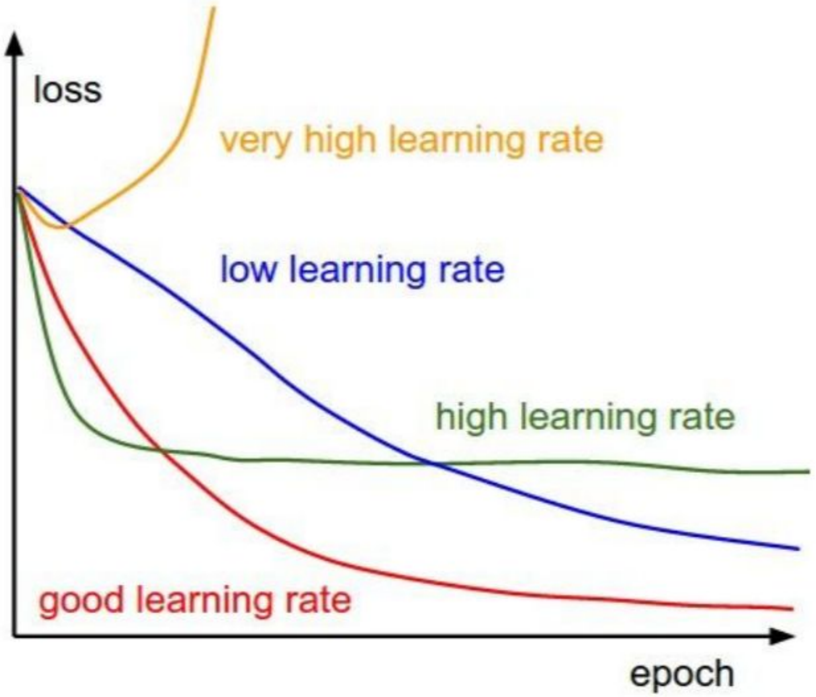
```
m = beta1*m + (1-beta1)*dx
v = beta2*v + (1-beta2)*(dx**2)
x += - learning_rate * m / (np.sqrt(v) + eps)
```

Hiperparâmetros

- Algoritmos de otimização
  - (mini batch) Stochastic Gradient Descent (SGD)
    - batch_size < len(dataset)
  - SGD com Momento
  - SGD com Momento de Nesterov
  - Adagrad
  - RMSProp
  - Adam (ada + m)
  - [...] (campo de pesquisa ativo)

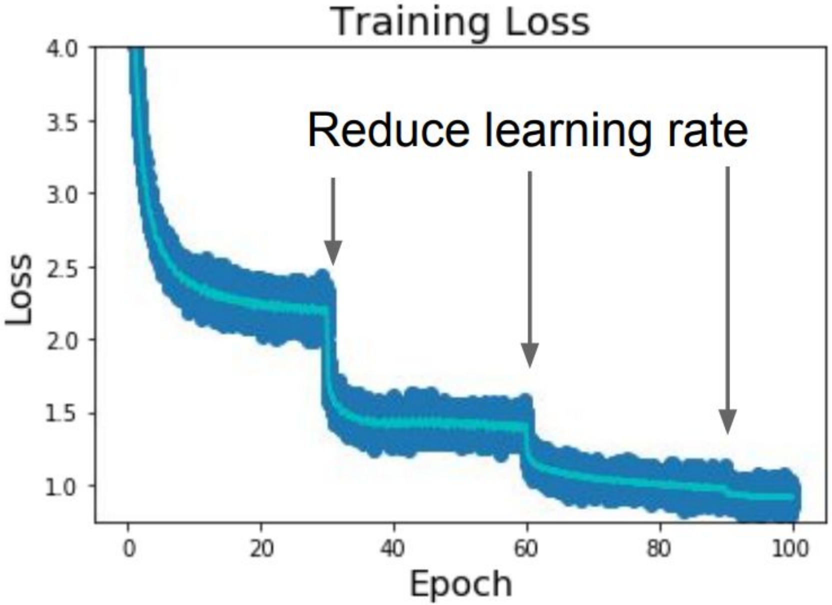
Hiperparâmetros

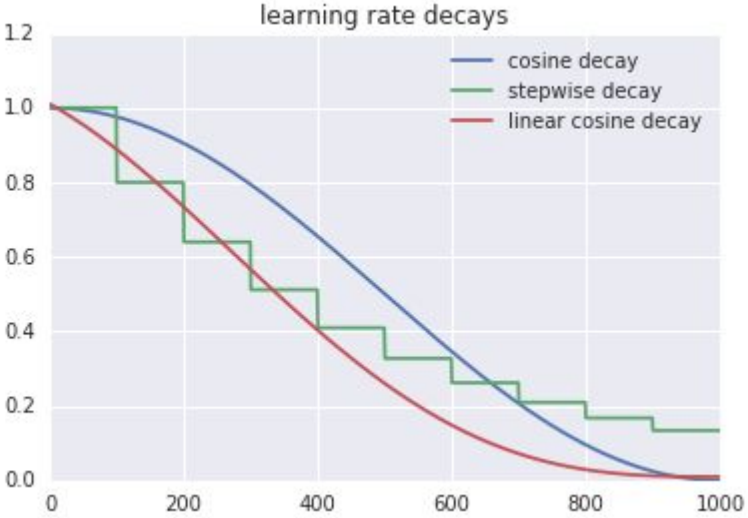
- Taxa de aprendizado (learning rate)



Hiperparâmetros

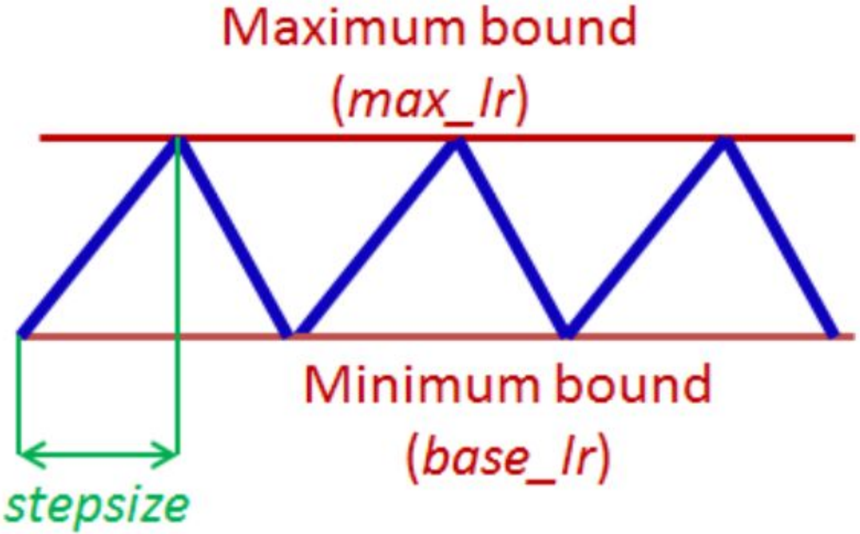
- Taxa de aprendizado (learning rate)
    - Decaimento

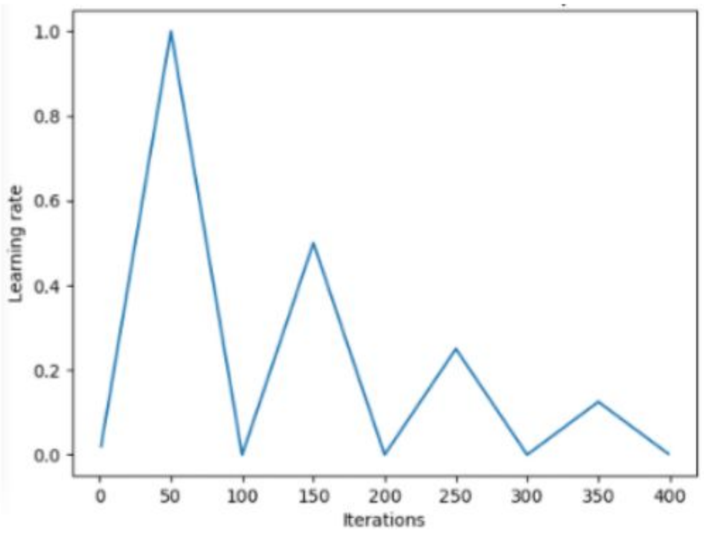




Hiperparâmetros

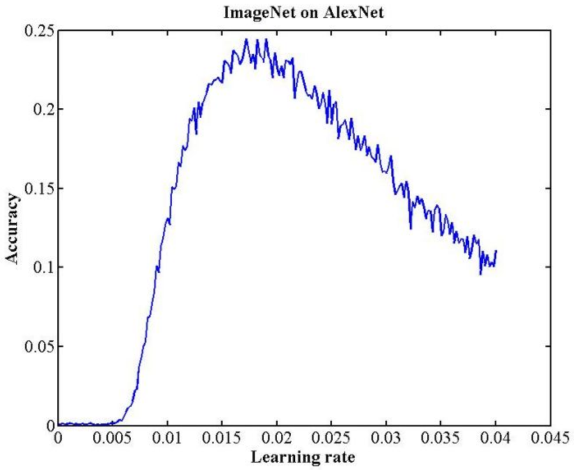
- Taxa de aprendizado (learning rate)
  - Decaimento
  - Taxas de aprendizado cíclicas (CLR)

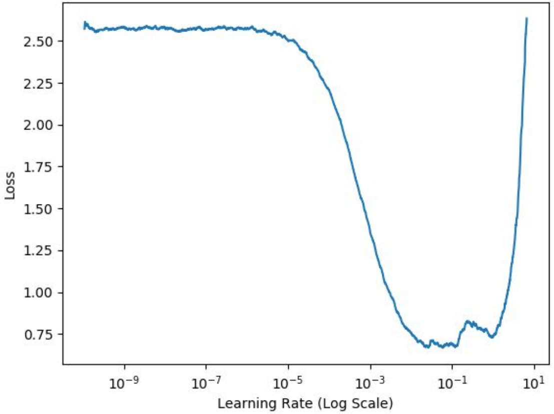




Hiperparâmetros

- Taxa de aprendizado (learning rate)
    - Decaimento
    - Taxas de aprendizado cíclicas (CLR)
    - Avaliação de LR (LR finder)





Hiperparâmetros

- Regularização
  - Penalização dos pesos

$$L = \underbrace{\frac{1}{N}\sum_i L_i}_{\text{data loss}} + \underbrace{\lambda R(W)}_{\text{regularization loss}}$$

- L2:

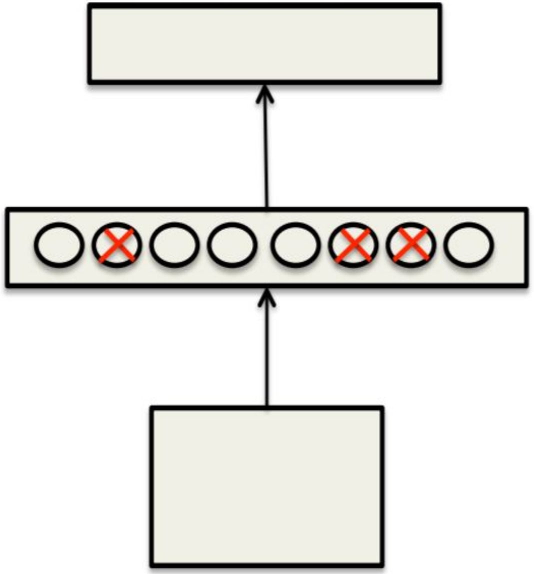
$$R(W) = \sum_k \sum_l W_{k,l}^2$$

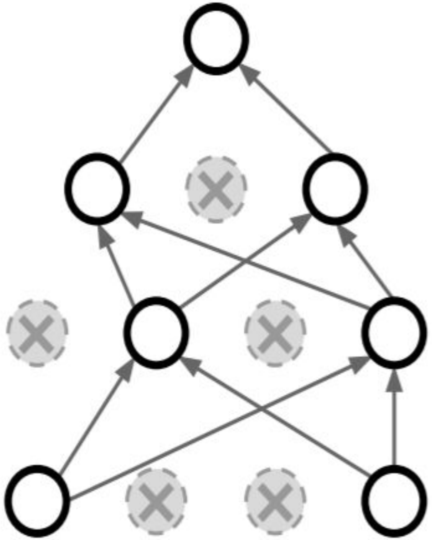
- L1:

$$R(W) = \sum_k \sum_l |W_{k,l}|$$

Hiperparâmetros

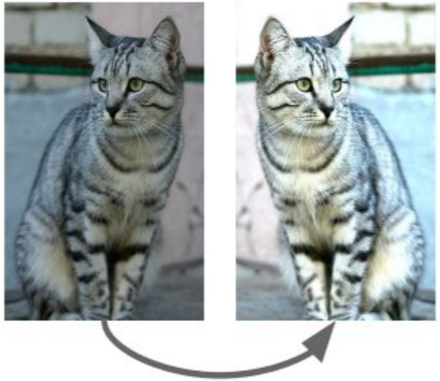
- Regularização
  - Dropout





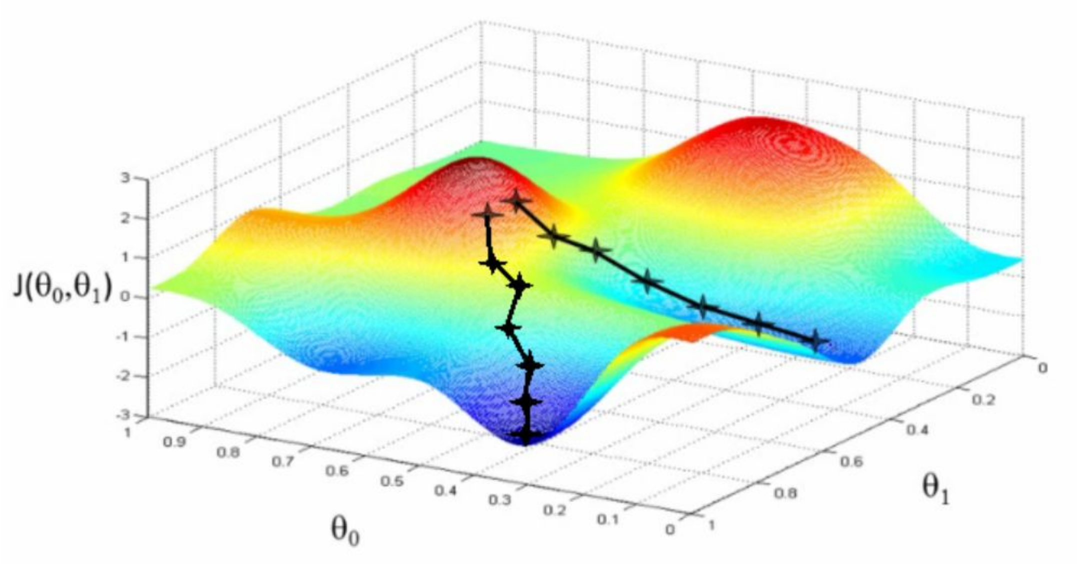
Hiperparâmetros

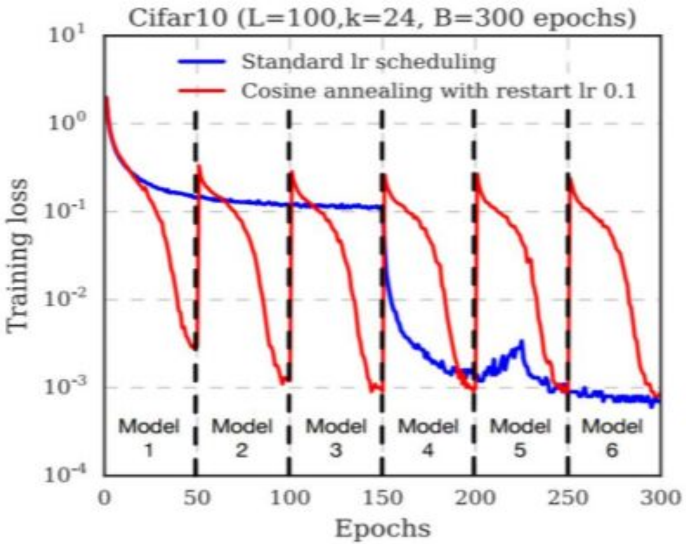
- Regularização
    - Data augmentation



Hiperparâmetros

- Regularização
  - Ensemble





Hiperparâmetros

- "Regularização"
    - BatchNorm

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

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

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

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

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

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

Hiperparâmetros

- Regularização = adiciona ruido no treino e marginaliza na inferência

Regularization: A common pattern

Training: Add random noise
Testing: Marginalize over the noise

Examples:
Dropout
Batch Normalization
Data Augmentation

Hiperparâmetros

- Regularização = adiciona ruido no treino e marginaliza na inferência

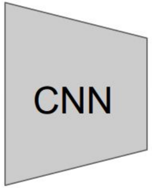
Regularization: Mixup

Training: Train on random blends of images

Testing: Use original images

Examples:

Dropout
Batch Normalization
Data Augmentation
DropConnect
Fractional Max Pooling
Stochastic Depth
Cutout
Mixup



Target label:
cat: 0.4
dog: 0.6

Randomly blend the pixels of pairs of training images, e.g. 40% cat, 60% dog

Zhang et al, "*mixup*: Beyond Empirical Risk Minimization", ICLR 2018

No próximo episódio...

- Transfer learning