Redes Neurais Artificiais

Professora: Anita Maria da Rocha Fernandes
Mestrando: Luiz Henrique A. Salazar

Agenda

- Treinamento da RNA
  - Função de Perda *(loss)*
  - Otimização
    - Fluxo de Treinamento
    - Descida do Gradiente *(Gradient Descent)*
    - Taxa de Aprendizado
  - Hiperparâmetros de Treinamento
    - Iteração
    - Batch
    - Época

Função de perda *(loss)*

https://youtu.be/rulM9AQOxDE

Função de perda *(loss)*

Aprendizado de uma rede como “tentativa e erro”?

- “Tentativa guiada”

Definição de uma medida numérica da qualidade :

- Objetivo de minimizar ou maximizar
- Patrick quer **minimizar** a distância entre a mão dele e a tampa.
    - Minimizar a distância entre um ponto e outro.

Função de perda *(loss)*

É também chamada de:

- Função **objetivo**
- Função de **perda**
- **Critério**
- ***Loss***
- entre outros.

Função de perda *(loss)*

*"A função de custo **reduz** todos os aspectos bons e ruins de um sistema complexo a um **único número**, um valor **escalar**, o que permite **rankear** e **comparar** as soluções candidatas."*

- Função de perda = único número
  - "Minimizar a **distância** e a **força** que o Patrick bate no pote"
    - Devemos "compactar" todas as métricas em um único valor escalar.

** Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks*

Função de perda *(loss)*

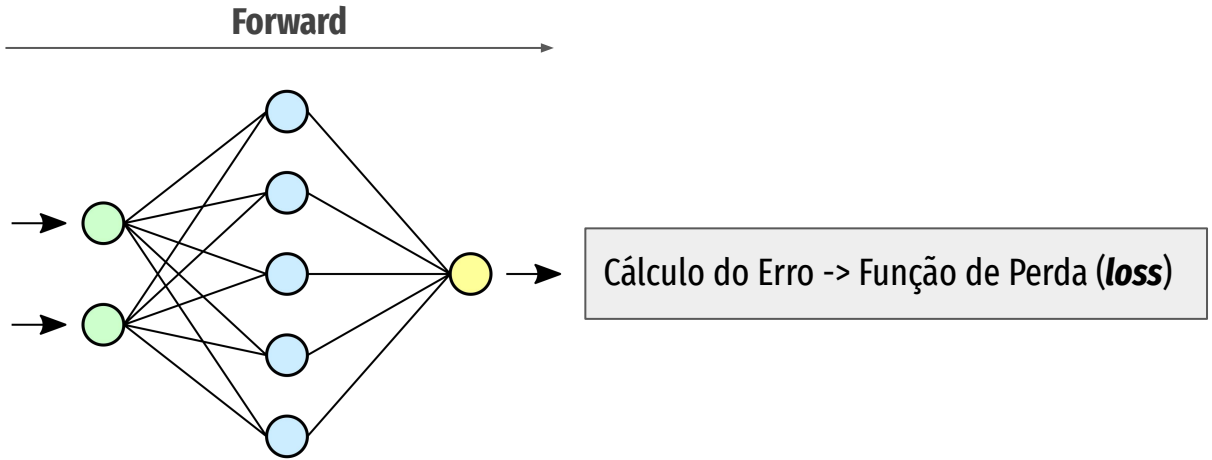
*"Se fizermos uma **escolha ruim** da função de custo e os **resultados** obtidos **não** forem **satisfatórios**, é nossa culpa por não especificar bem nosso objetivo"*

- RNA não converge de jeito nenhum!
  - Faltou **treinamento**?
  - Escolha da rede **não é a ideal**?
  - Pode ser necessário **melhorar a função de perda**!
    - Está relacionada diretamente com o problema.

** Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks*

Função de perda *(loss)*

Ao treinar o modelo, o objetivo é fazê-lo **convergir** para uma solução aproximadamente ótima.
A função de perda é utilizada para acompanhar essa **convergência**.



Função de perda *(loss)*

As funções de perda são divididas em funções de *loss* de **regressão** e de **classificação**.

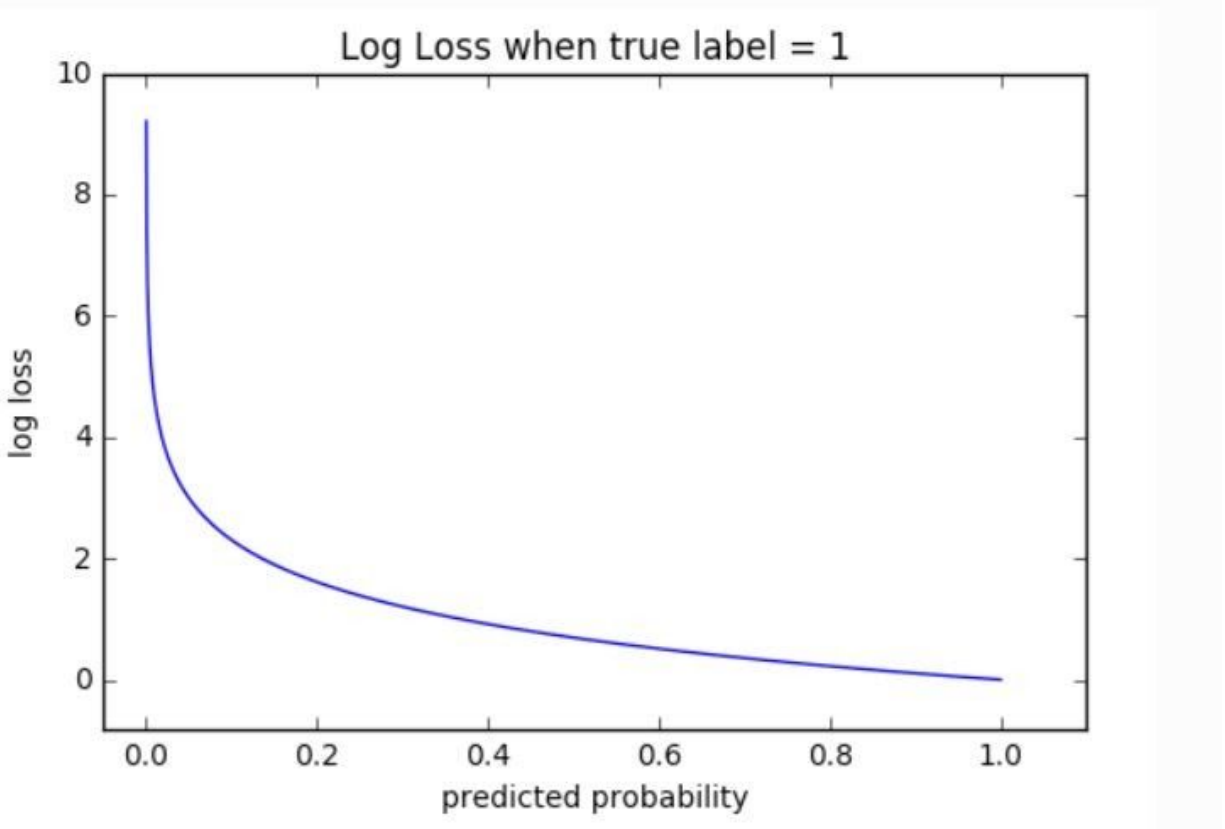
- **Regressão**:
  - Erro Médio Quadrático (***M**ean **S**quare **E**rror* — MSE)
    - Computa a média quadrática da **diferença** entre o valor **real** e o valor **predito**.
  - Erro Médio Absoluto (**M**ean **A**bsolute **E**rror — MAE)
    - Utiliza o **módulo** de cada erro e mede apenas a **distância do valor real**, independente de ser acima ou abaixo.

Função de perda *(loss)*

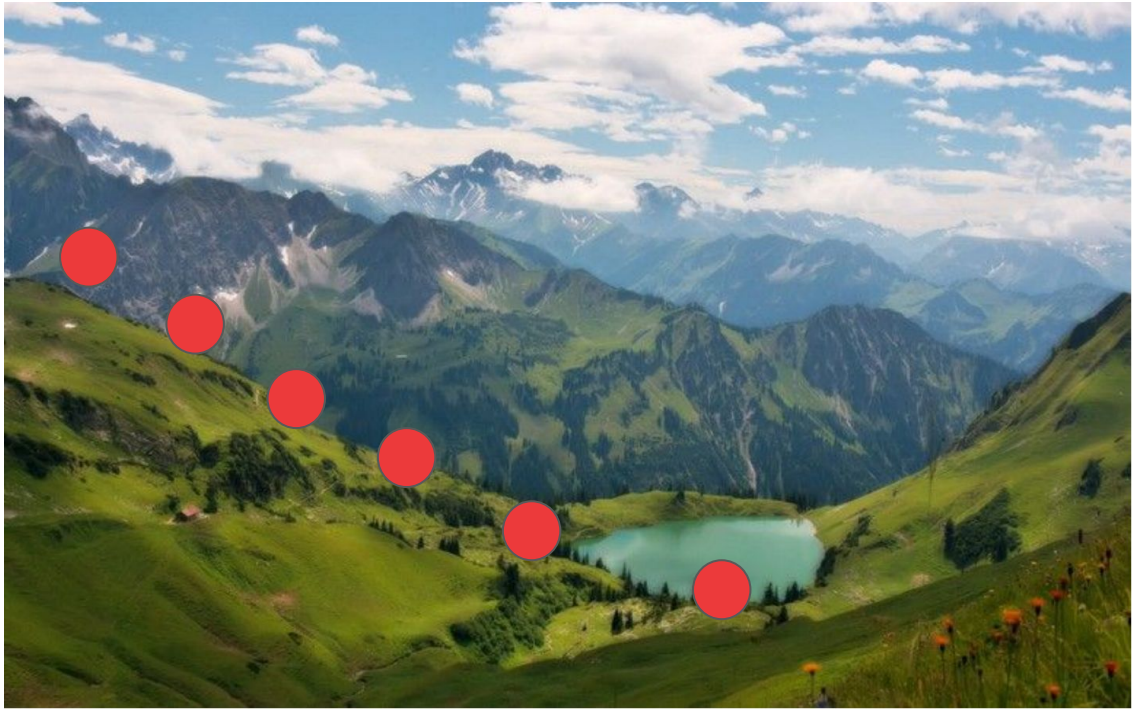
As funções de perda são divididas em funções de *loss* de **regressão** e de **classificação**.

- **Classificação (Binária)**:
  - Entropia Cruzada (Cross-Entropy Loss - **Log Loss**)
    - Retorna um valores entre 0 e 1
    - Mais próximo de **zero (0)**:
      - maior a **probabilidade** da classificação ser a **correta**.
    - Mais próximo de **um (1)**:
      - maior a **probabilidade** de que o modelo está **errando**

Função de perda *(loss)*

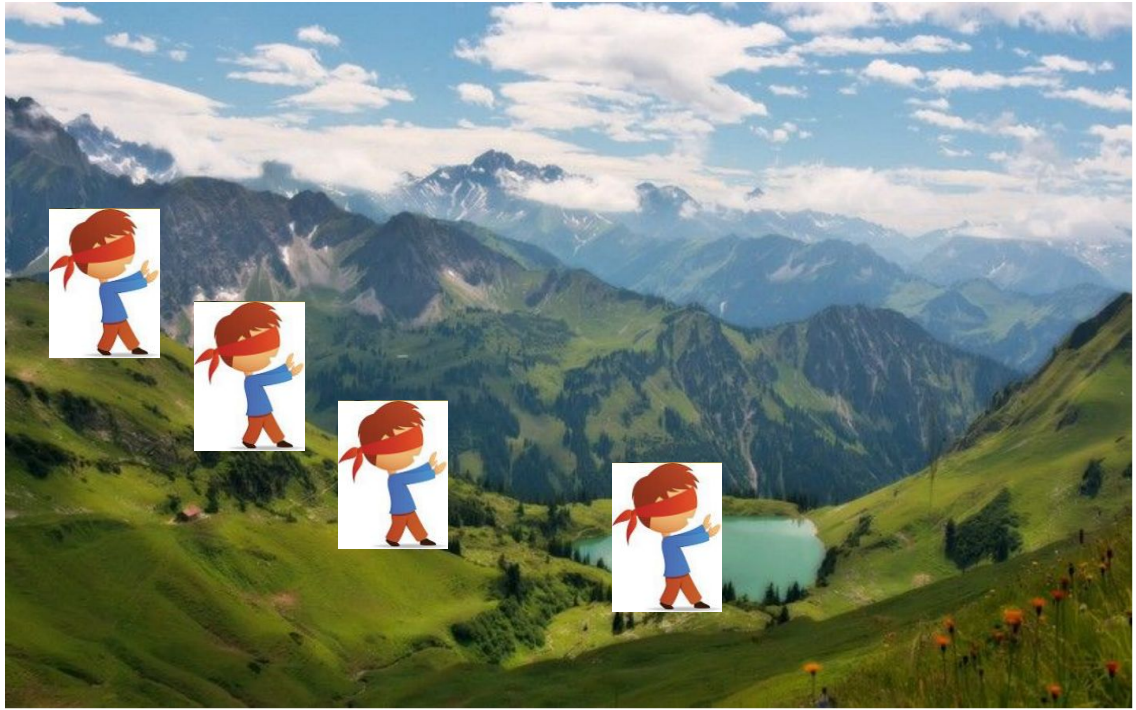


Otimização



http://cs231n.stanford.edu/slides/2016/winter1516_lecture3.pdf

Otimização

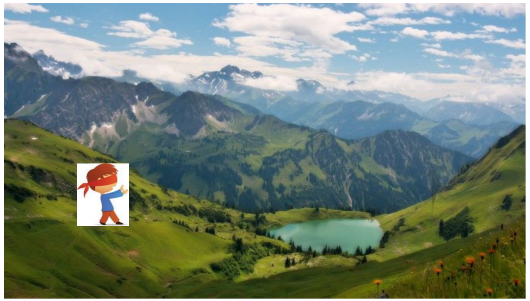


http://cs231n.stanford.edu/slides/2016/winter1516_lecture3.pdf

Otimização

Objetivo é **minimizar** a função de perda.

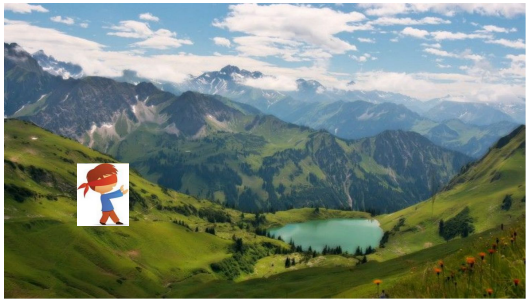
- Altera-se os pesos do modelo **iterativamente** e verifica-se o impacto de cada alteração na função de perda.



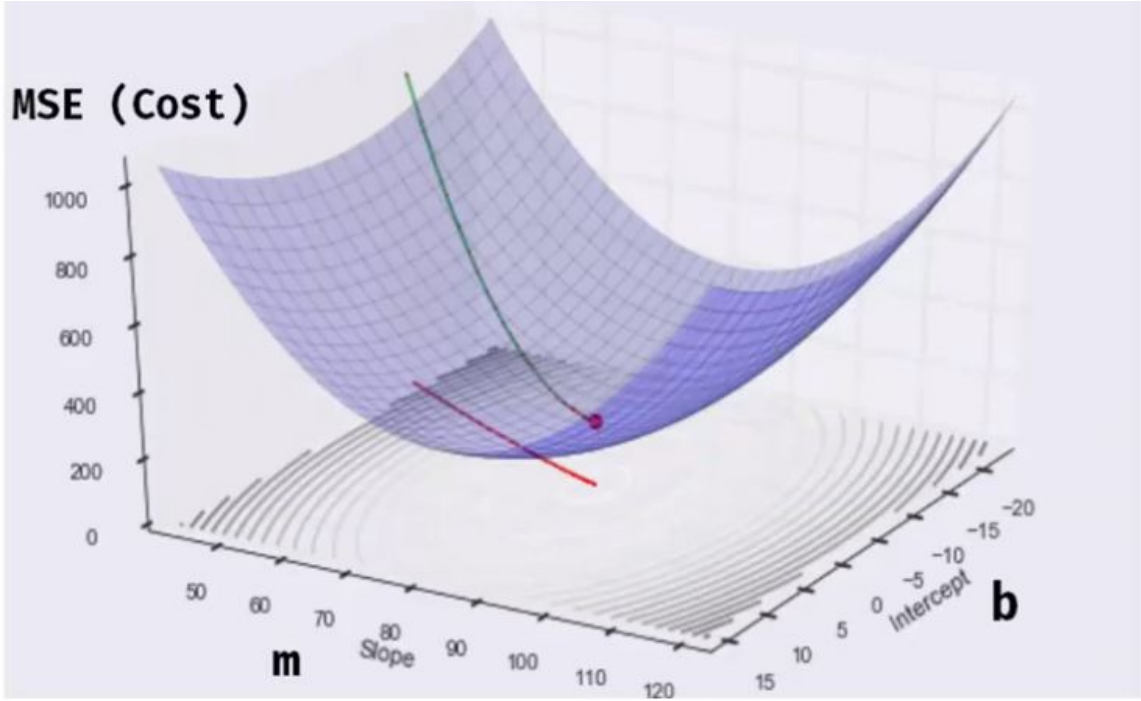
Otimização

Objetivo é **minimizar** a função de perda

- Um “passo” é uma alteração de **pesos**
- A “altura” é a função de **perda** em relação aos pesos.
- A montanha é a superfície de **erro**



Otimização

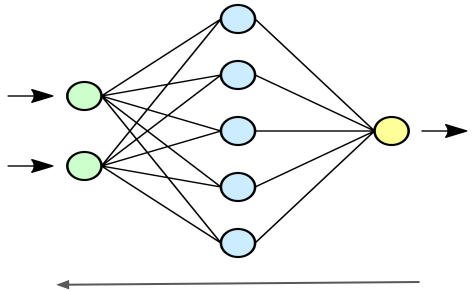


https://www.infinitycodex.in/data-science-ss-106gradient-descent-and

Otimização

Para calcular se descemos ou subimos a montanha após um passo, calculamos a **derivada** dessa função.

- Derivada entre dois pontos que indicará numa reta a direção que foi seguida após alterar os pesos.
  - Indica **quanto** e
  - em qual **direção** o menino caminhou.



Backpropagation

Otimização

Para múltiplas dimensões, o vetor de derivadas parciais é chamado de **gradiente.**

Logo, o **gradiente** é o indicador se um **passo** (alteração nos pesos da rede) dado na montanha, **melhorou** ou piorou o **modelo**, com base em uma **medida de qualidade** (função de perda).

Otimização

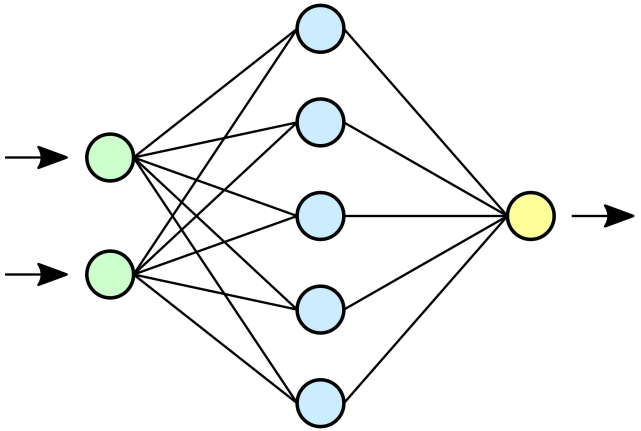
A **otimização** orienta a escolha dos próximos passos com base no valor obtido no **gradiente**.

Representa o uso da informação para **escolher** a próxima **alteração de pesos** da RNA, que talvez leve ao **ponto mínimo** da superfície de erro (“vale da montanha”)

Fluxo de Treinamento

O fluxo de treinamento se resume nos seguintes passos iterativos:

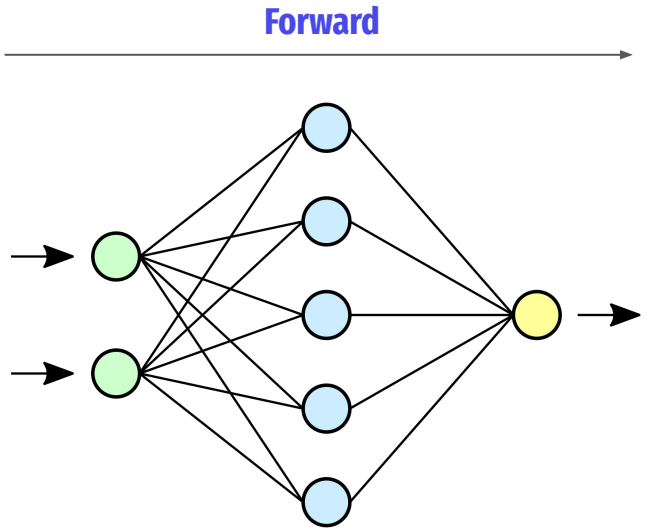
1. Entrada da rede
2. Cálculo da função de perda
3. Cálculo do gradiente
4. Atualização dos pesos
5. Volta para o passo 1



Fluxo de Treinamento

O fluxo de treinamento se resume nos seguintes passos iterativos:

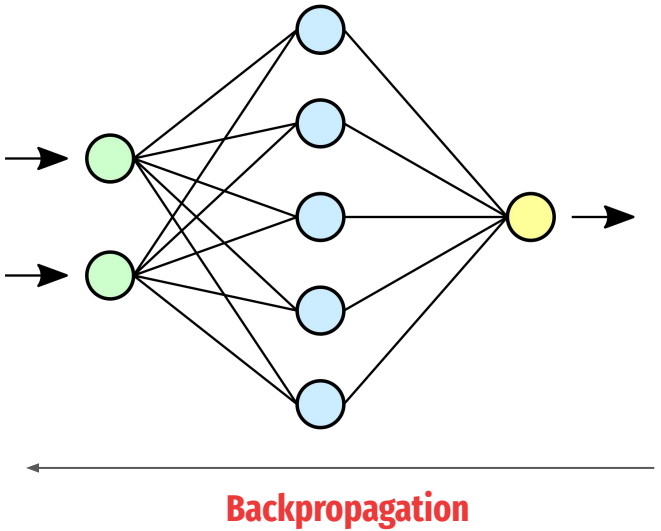
1. **Entrada da rede**
2. **Cálculo da função de perda**
3. Cálculo do gradiente
4. Atualização dos pesos
5. Volta para o passo 1



Fluxo de Treinamento

O fluxo de treinamento se resume nos seguintes passos iterativos:

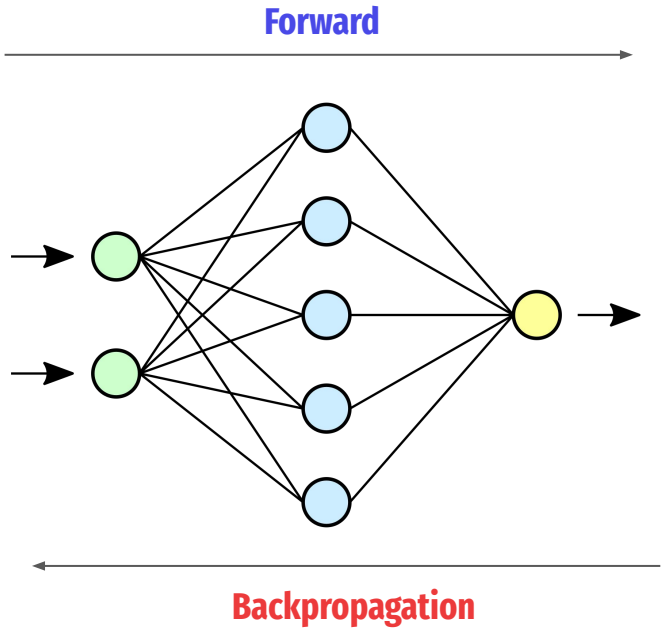
1. Entrada da rede
2. Cálculo da função de perda
3. **Cálculo do gradiente (∇)**
4. **Atualização dos pesos**
5. Volta para o passo 1



Fluxo de Treinamento

O fluxo de treinamento se resume nos seguintes passos iterativos:

1. Entrada da rede
2. Cálculo da função de perda
3. Cálculo do gradiente
4. Atualização dos pesos
5. **Volta para o passo 1**



Descida do Gradiente (*Gradient Descent*)

Descida do Gradiente é o **algoritmo** clássico de otimização.

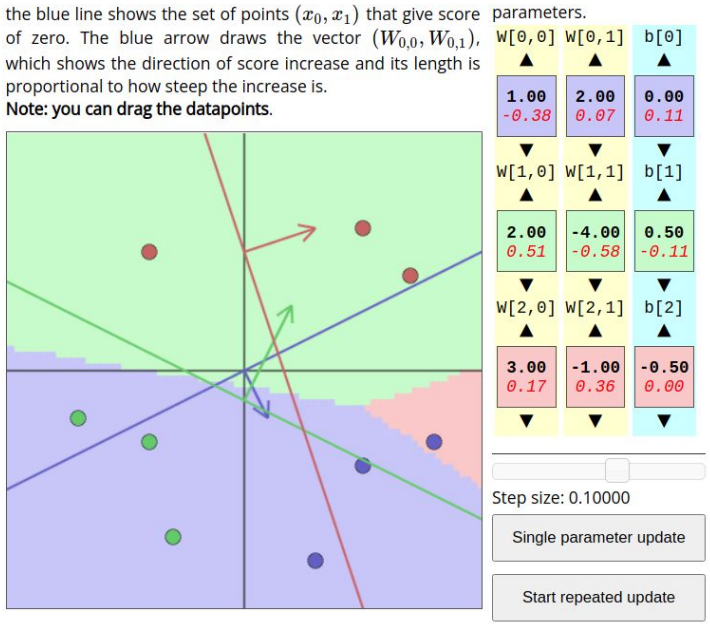
- Consiste em subtrair o valor do gradiente (∇f) dos pesos (W) da rede.

$$W_i = W_i - \alpha * \nabla f$$

- O tamanho do passo de otimização é controlado pelo multiplicador α -> **Taxa de Aprendizado**

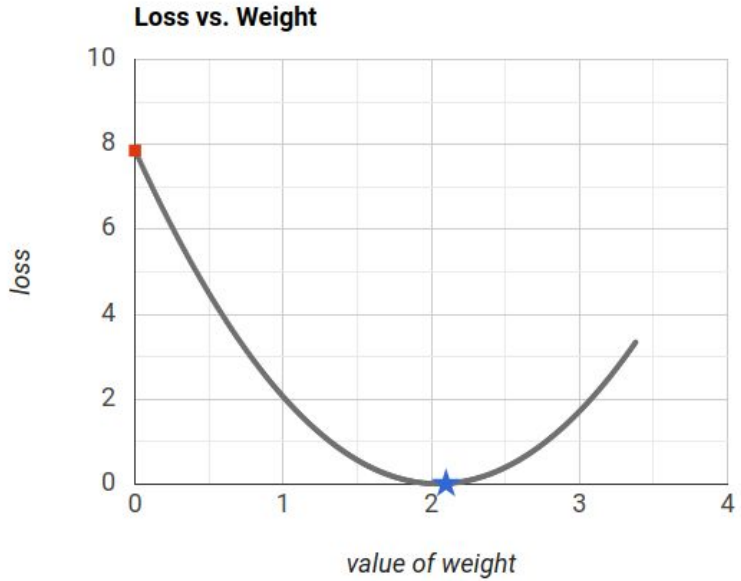
Taxa de Aprendizado

Qual o efeito de aumentar ou diminuir a taxa de aprendizado?

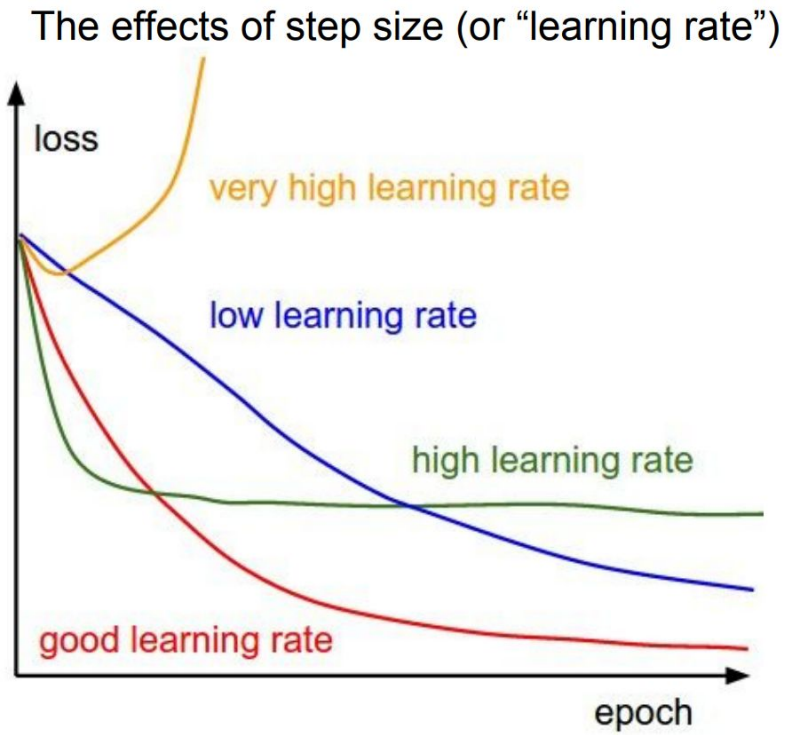


Taxa de Aprendizado

Qual o efeito de aumentar ou diminuir a taxa de aprendizado?



Otimização


The effects of step size (or “learning rate”)
loss
very high learning rate
low learning rate
high learning rate
good learning rate
epoch

Aula 3 - Redes Neurais - Loss e Otimização

Hiperparâmetros de Treinamento

3 etapas importantes no processo de treinamento de uma RNA:

- Organizam o treinamento da rede

Iteração

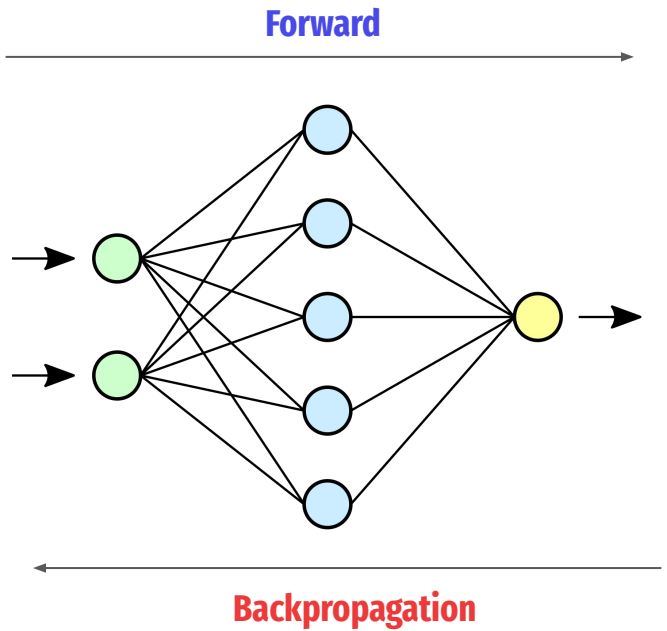
Batch

Época

Iteração

O fluxo de treinamento se resume nos seguintes passos **iterativos**:

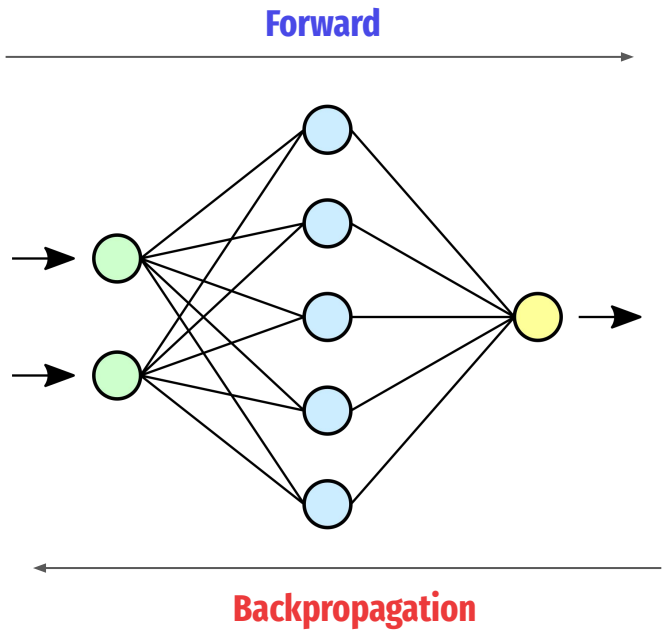
1. Entrada da rede
2. Cálculo da função de perda
3. Cálculo do gradiente
4. Atualização dos pesos
5. **Volta para o passo 1**



Iteração

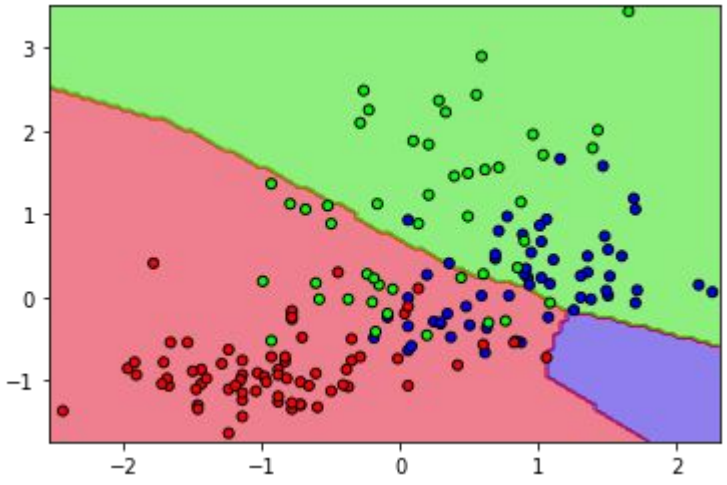
Uma iteração consiste em um **passo de otimização**:

Forward + **Backpropagation**

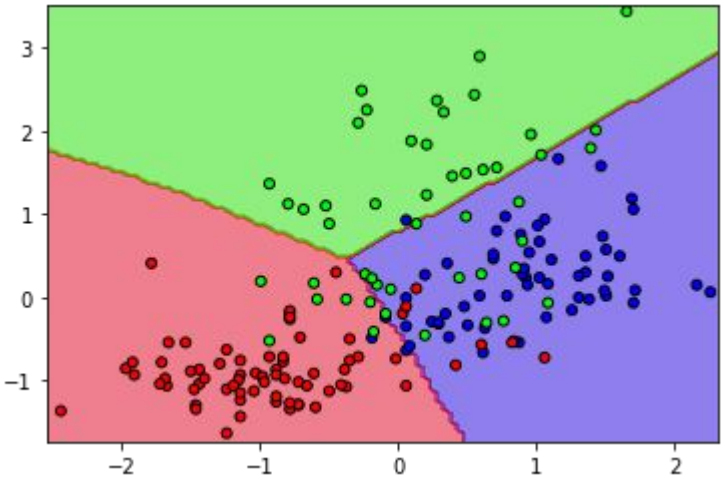


Iteração

10 **iterações**



100 **iterações**



Batch

É a **quantidade de amostras** vistas numa iteração, ou seja, em um passo de treinamento (uma iteração).

“Aumento o tamanho de amostras e consigo otimizar o modelo com menos iterações?”

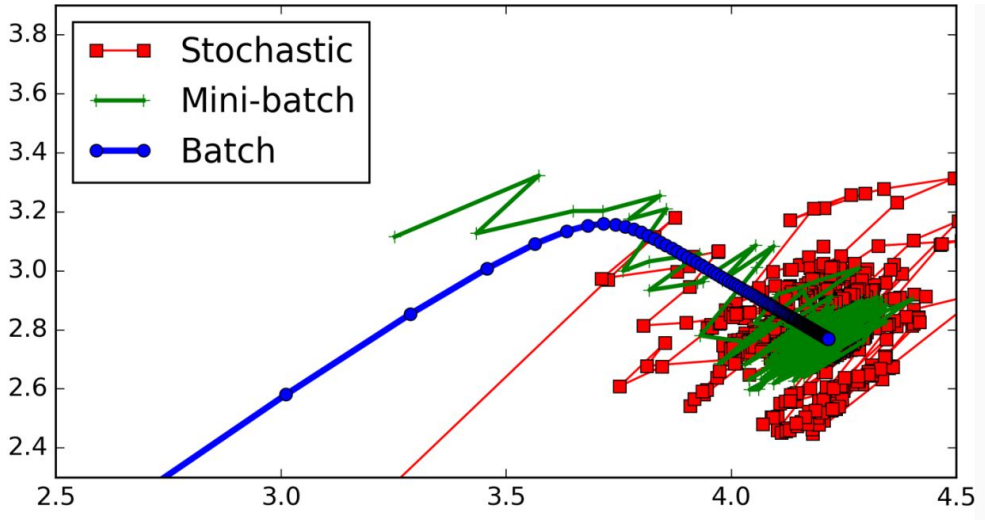
Batch

O tamanho do batch interfere no comportamento de **convergência**.

Estocástico: uma amostra por vez (mais **rápido**)

Mini-batch: subconjunto do treino

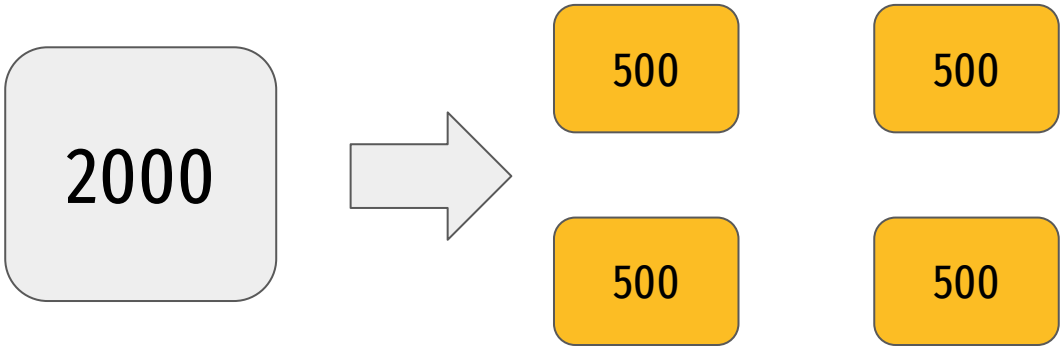
Batch: conjunto de treino completo (todas as amostras de treino) (mais **demorado**)



Batch

Se conjunto de dados é **pequeno** (< 3000), use **todos** os dados

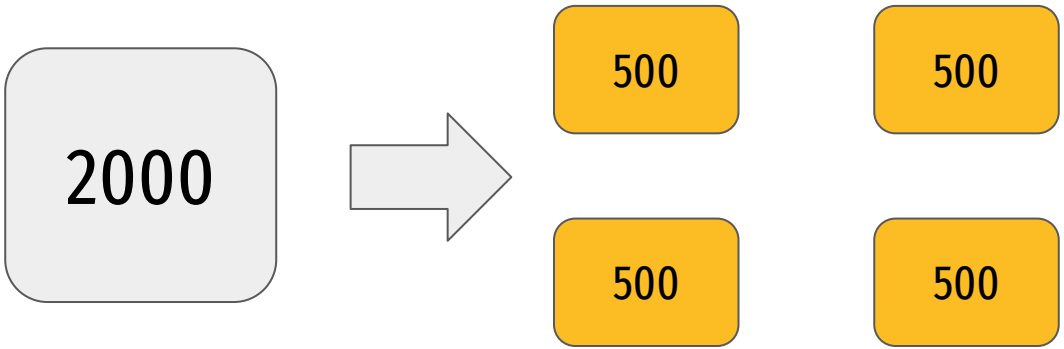
- 2000 amostras de treino e 1000 para teste
  - Batches de tamanho 500.



Batch

Para que a rede veja todo o conjunto de treino, são necessárias 4 iterações.

- Completando uma época!



Época

Quando **todas as amostras** do conjunto de treino foram vistas pelo modelo, completo uma **época.**

Pergunta: Se ao final de uma época o modelo **já viu todas as amostras** de treino, por que precisamos de **mais uma época**?

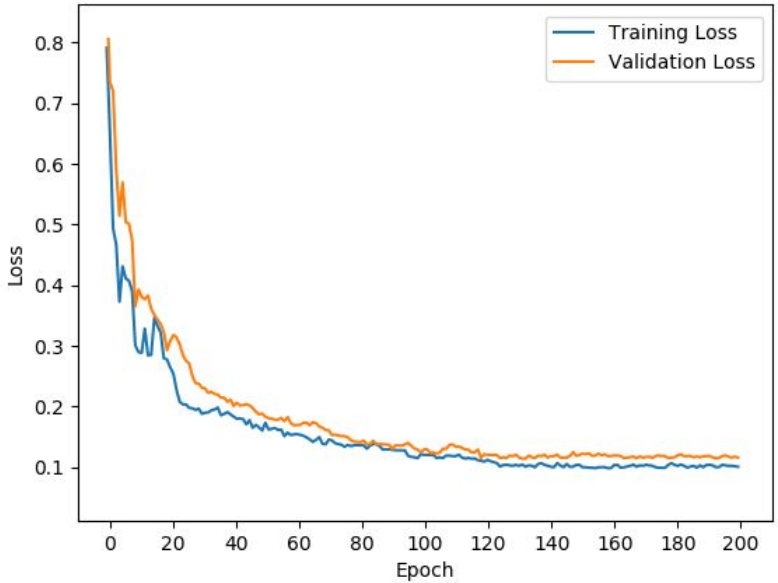
Época

Pergunta: Se ao final de uma época o modelo **já viu todas as amostras** de treino, por que precisamos de **mais uma época**?

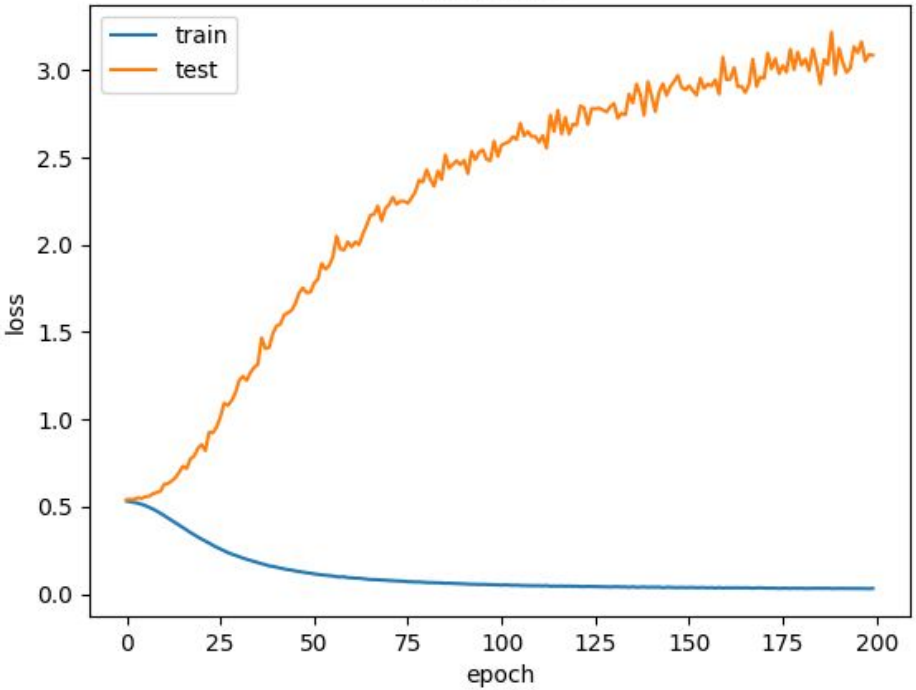
- O treinamento (otimização) é um processo **iterativo** e possui pequenos ajustes a partir do modelo inicial.
- A época seguinte (n+1) tem como ponto de partida o modelo **ajustado** na época anterior (n).

Época

Em geral, o **gráfico de convergência** é definido em termo das épocas, comparado com a *loss* obtida.



Época


train
test
loss
epoch
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
25
50
75
100
125
150
175
200

Demo Deep Learning Absenteísmo