

INTELIGÊNCIA ARTIFICIAL

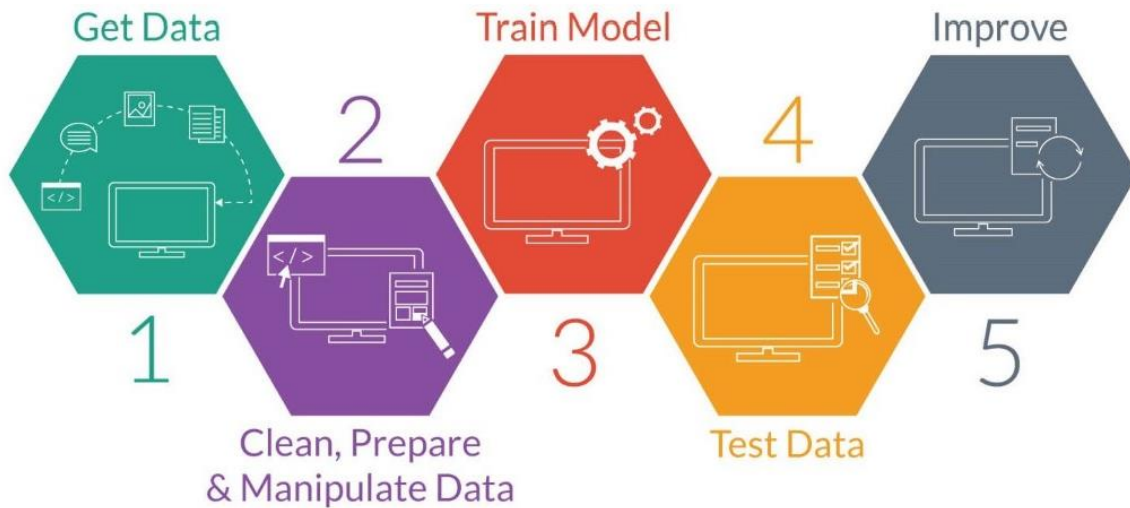
REDES NEURAIS ARTIFICIAIS



Krigor R. Rosa da Silva
krigor.silva@edu.udesc.br



Workflow ML





Tipos de Aprendizado

● Aprendizado Supervisionado

- Treinamento através de um conjunto de dados onde para cada entrada a saída é conhecida
- Regressão e classificação

● Aprendizado Não supervisionado

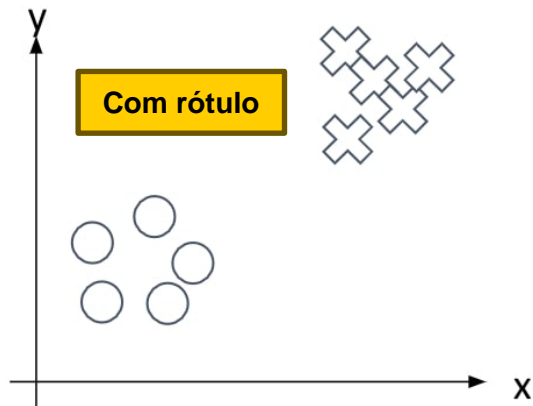
- Dados sem rótulo, a saída não é informada
- Agrupamento, associação

● Aprendizado por Reforço

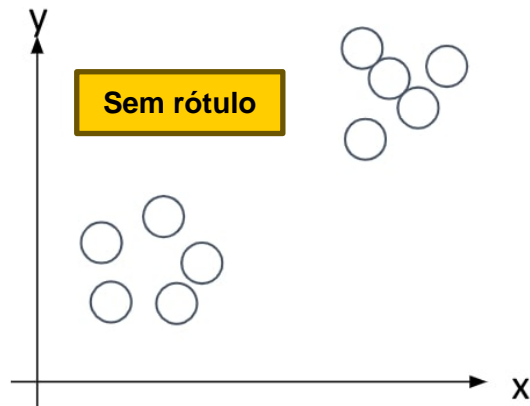
- O agente aprende por tentativa e erro usando feedback para suas próprias ações



Tipos de Aprendizado



- Qual será o valor ou classe previsto para um novo dado?



- Meus dados podem ser divididos em grupos?



Aprendizado Supervisionado

● Dados rotulados

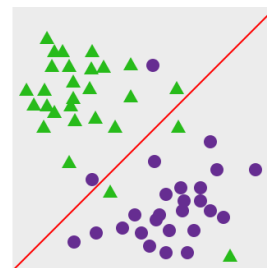
● Regressão

- Saída desejada é contínua
- Obter uma função que se aproxime da verdadeira (desconhecida)



● Classificação

- Saída desejada é discreta ou categórica





Base de dados

- O conjunto de dados contém informações para a criação do nosso modelo
- Formado por variáveis independentes e dependentes
- Podem ser contínuas, binárias ou categóricas

iris setosa



petal

sepal

iris versicolor



petal

sepal

iris virginica



petal

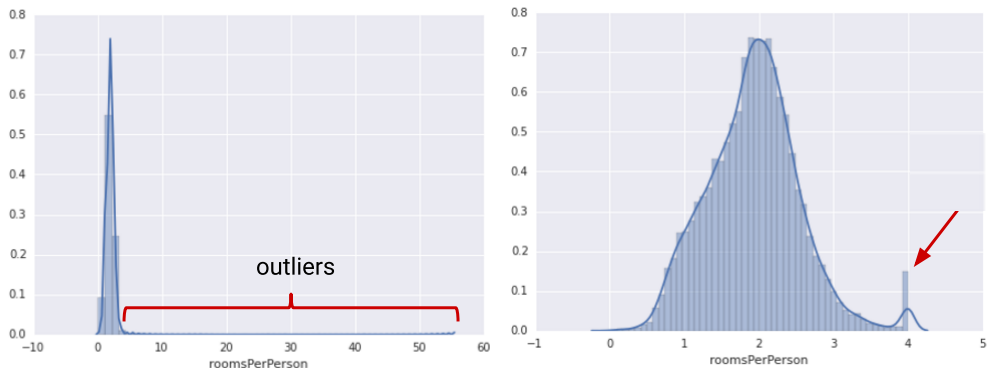
sepal

sepal_length	sepal_width	petal_length	petal_width	class
5.1	3.5	1.4	0.2	iris_setosa
4.9	3	1.4	0.2	iris_setosa
4.6	3.2	1.4	0.2	iris_setosa
5.3	3.7	1.5	0.2	iris_setosa
5	3.3	1.4	0.2	iris_setosa
7	3.2	4.7	1.4	iris_versicolor
6.5	2.8	4.6	1.5	iris_versicolor
5.7	2.8	4.5	1.3	iris_versicolor
6.3	3.3	4.7	1.6	iris_versicolor
4.9	2.4	3.3	1	iris_versicolor
6.3	3.3	6	2.5	iris_virginica
5.8	2.7	5.1	1.9	iris_virginica
7.1	3	5.9	2.1	iris_virginica
6.3	2.9	5.6	1.8	iris_virginica
6.5	3	5.8	2.2	iris_virginica



Base de dados

Limpeza dos dados



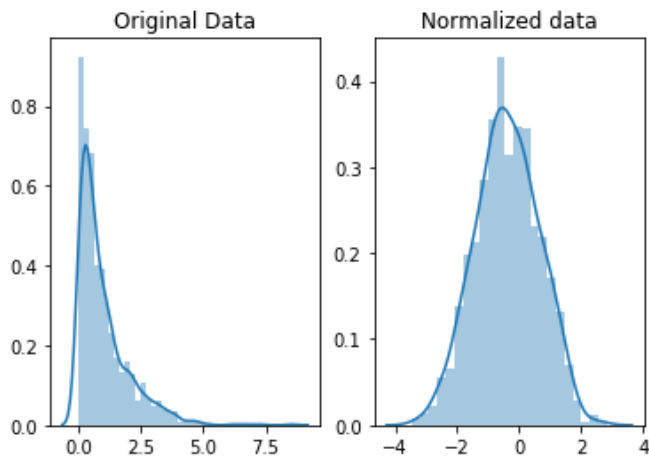
- ⦿ Dados incompletos
- ⦿ Informações erradas
- ⦿ Outliers

Todo outlier deve ser removido?



Base de dados

Normalização dos dados



1.

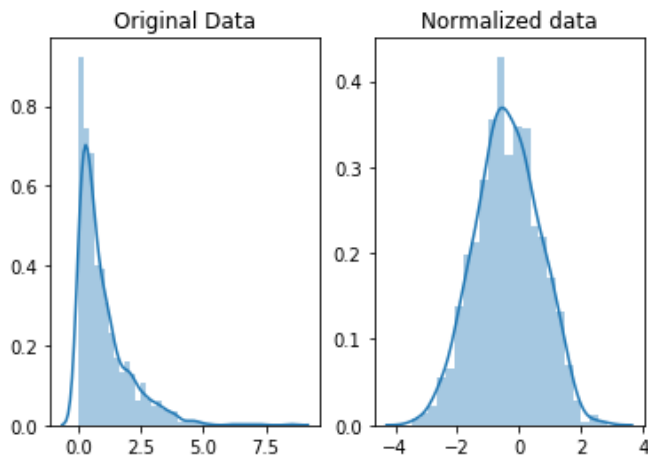
2.

3.



Base de dados

Normalização dos dados



1. Convergência mais rápida
2. Evitar vies
3. Redução do impacto de outliers



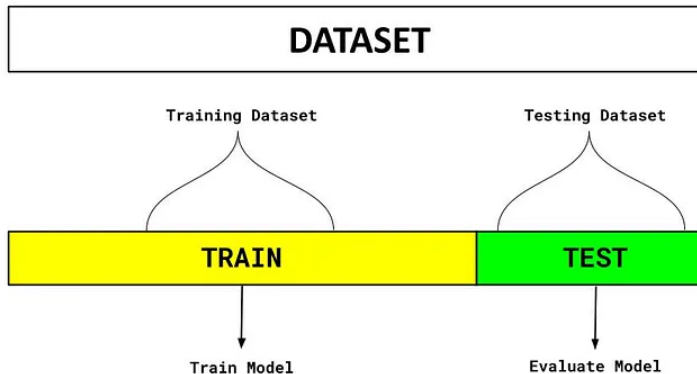
Base de dados

- Fases de Aprendizagem
 - Treinamento
 - Teste
- Separação dos conjuntos de dados para treinar uma rede neural artificial;
 - Holdout
 - K-fold Cross-Validation
 - Leave-One-Out Cross-Validation



Base de dados

- Método Holdout
- Separa um conjunto de dados em dois conjuntos: um conjunto de treinamento e um conjunto de teste
- 2/3 para treinamento e 1/3 para teste





Base de dados

- Método K-fold Cross-Validation
- Divide o conjunto de dados em K partes iguais, treina o modelo K vezes
- Quais as vantagens em relação ao método holdout?

Split 1	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Metric 1
Split 2	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Metric 2
Split 3	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Metric 3
Split 4	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Metric 4
Split 5	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Metric 5

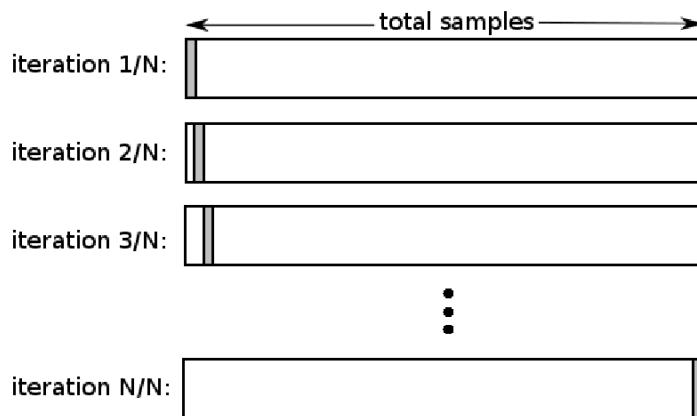
Training data

Test data



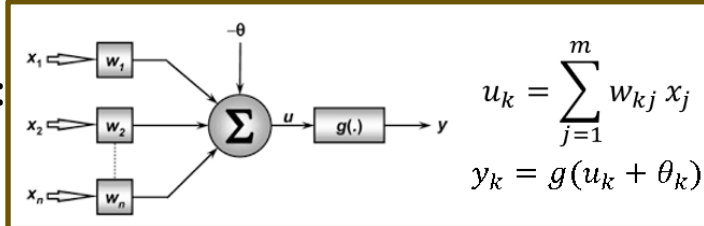
Base de dados

- Método Leave-One-Out Cross-Validation
- Cada exemplo de dados é usado como conjunto de validação uma vez, enquanto o restante é usado para treinamento
- Quando a base de dados é muito pequena





Algoritmo de Aprendizado: Perceptron



1. Pesos são inicializados aleatoriamente com valores pequenos
2. Para cada exemplo de treinamento T:
 - Cálculo da saída da rede $y(T)$
 - Se a saída calculada for diferente do valor desejado $d(T)$
 - Ajuste dos pesos e bias: adiciona-se um delta

$$\Delta w_{kj} = \eta * (d(T) - y(T)) * x_j$$

$$\Delta \theta_j = \eta * (d(T) - y(T))$$

$$w_{kj \text{ novo}} = w_{kj \text{ velho}} + \Delta w_{kj}$$

$$\theta_k \text{ novo} = \theta_k \text{ velho} + \Delta \theta_k$$

- Processo é feito para todos os pesos da rede
- Vai para a próxima iteração ou época



Algoritmo de Aprendizado: Taxa de Aprendizado

- Parâmetro η é a taxa de aprendizado
- Geralmente um valor escolhido entre 0 e 1
- Controla o movimento dos pesos
 - Muito pequeno:
 - Muito grande:



Algoritmo de Aprendizado: Taxa de Aprendizado

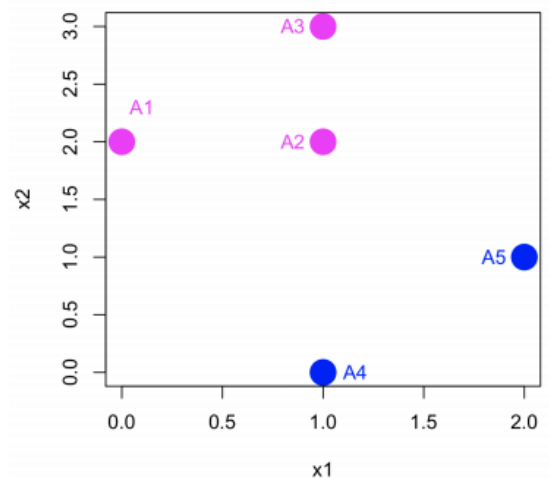
- Parâmetro η é a taxa de aprendizado
- Geralmente um valor escolhido entre 0 e 1
- Controla o movimento dos pesos
 - Muito pequeno: Uma taxa pequena tornará o aprendizado mais estável, porém em um processo mais lento;
 - Muito grande: Uma taxa alta pode resultar em atualizações oscilatórias, podendo não obter padrões eficientes e não encontrar a melhor solução, porém o processo se torna mais rápido;



Exemplo

Problema de classificação

PONTOS	RESPOSTA DESEJADA
$A1=(0,2)$	1
$A2=(1,2)$	1
$A3=(1,3)$	1
$A4=(1,0)$	0
$A5=(2,1)$	0

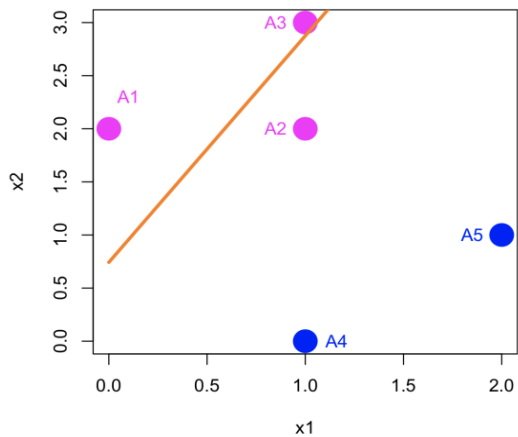




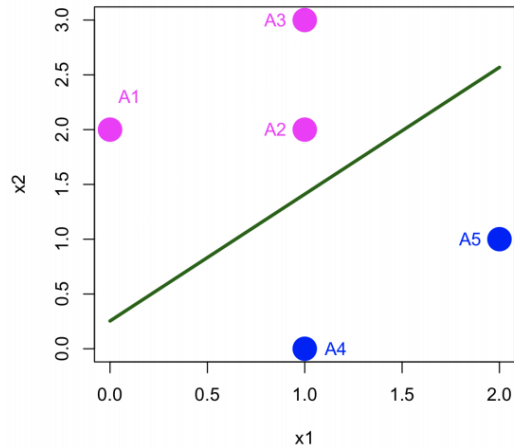
Exemplo

Problema de classificação

Iteração 1



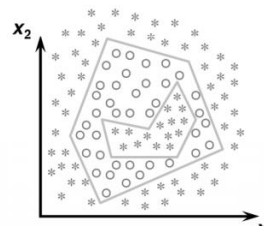
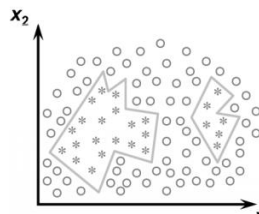
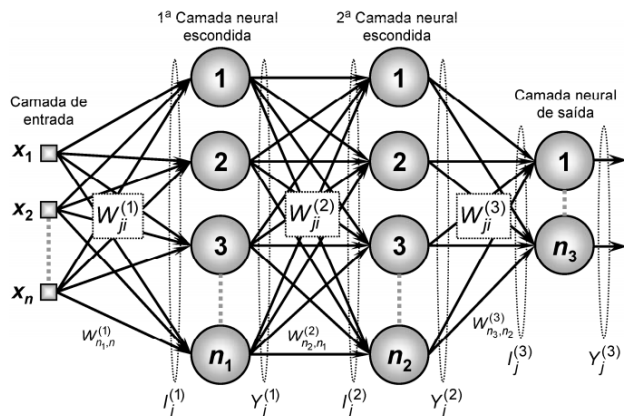
Iteração Final





Multilayer Perceptron (MLP)

- Redes Perceptron não são capazes de aprender conceitos complexos, porém, formam a base para a construção de um tipo de rede que pode aprender conceitos mais sofisticados
- Neurônios entre camadas vizinhas estão completamente conectados
- A regra de aprendizado no MLP baseia-se em cálculo diferencial

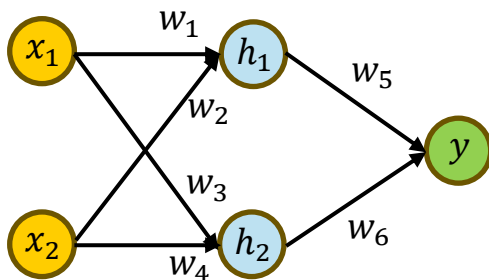




Backpropagation

- Algoritmo de treinamento para redes neurais que permite o ajuste dos pesos

Propagação direta



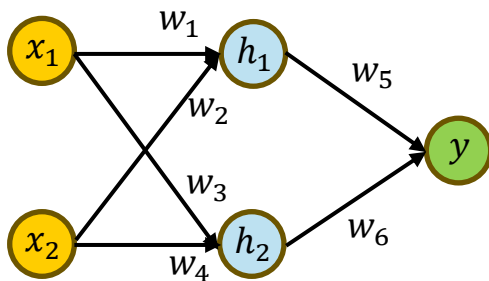
$$Erro = \frac{1}{2} (predição - real)^2$$

- Entradas passam pela rede gerando uma previsão



Backpropagation

Propagação inversa



$$Erro = \frac{1}{2} (predição - real)^2$$

- A propagação inversa é feita em cada camada até o início da rede, gerando os deltas e atualizando os pesos



Backpropagation

- Deseja-se minimizar o valor da função de perda da rede (Erro)
- Problema de otimização

$$y = g(x_1w_1 + x_2w_2 + x_3w_3 + \theta_1) \longrightarrow \text{Perceptron}$$



Backpropagation

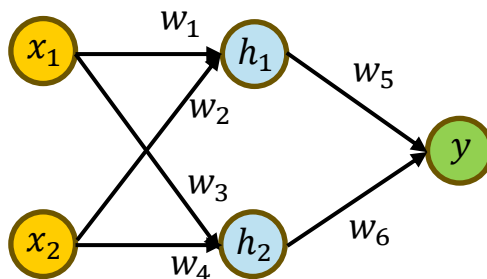
- Deseja-se minimizar o valor da função de perda da rede (Erro)
- Problema de otimização

$$y = g(x_1w_1 + x_2w_2 + x_3w_3 + \theta_1) \longrightarrow \text{Perceptron}$$

$$y = g((h_1)w_5 + (h_2)w_6 + \theta_2)$$

$$y = g(g(x_1w_1 + x_2w_2 + \theta_1))w_5 + g((x_1w_3 + x_2w_4 + \theta_1))w_6 + \theta_2)$$

MLP





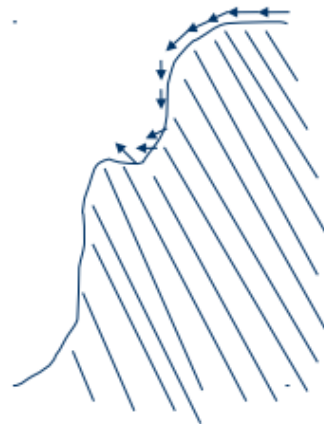
Backpropagation

- O backpropagation implementa uma busca por descida do gradiente (gradient descent) através do espaço dos possíveis pesos da rede.
- É usado para encontrar a direção em que o erro decresce mais rapidamente.

$$\text{gradiente} = \left(\frac{\partial \text{Erro}}{\partial w_j} \right)$$

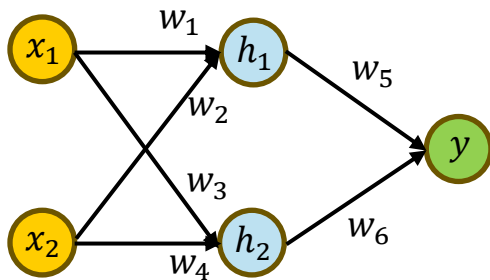
$$w_{kj \text{ novo}} = w_{kj \text{ velho}} - \eta \left(\frac{\partial \text{Erro}}{\partial w_j} \right) x_j$$

$$\theta_j \text{ novo} = \theta_j \text{ velho} - \eta \left(\frac{\partial \text{Erro}}{\partial w_j} \right)$$





Backpropagation



$$w_{6 \text{ novo}} = w_{6 \text{ velho}} - \eta \left(\frac{\partial \text{Erro}}{\partial w_6} \right) x_j$$

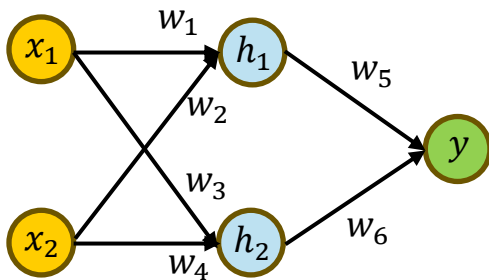
$$\frac{\partial \text{Erro}}{\partial w_6} = \frac{\partial \text{Erro}}{\partial y} * \frac{\partial y}{\partial w_6}$$

$$y = g(g((x_1 w_1 + x_2 w_2 + \theta_1)) w_5 + g((x_1 w_3 + x_2 w_4 + \theta_1)) w_6 + \theta_2)$$

$$\text{Erro} = \frac{1}{2} (y - \text{real})^2$$



Backpropagation



$$w_{6\text{ novo}} = w_6 - \eta \left(\frac{\partial \text{Erro}}{\partial w_6} \right)$$

$$\frac{\partial \text{Erro}}{\partial w_6} = \frac{\partial \text{Erro}}{\partial y} * \frac{\partial y}{\partial w_6}$$

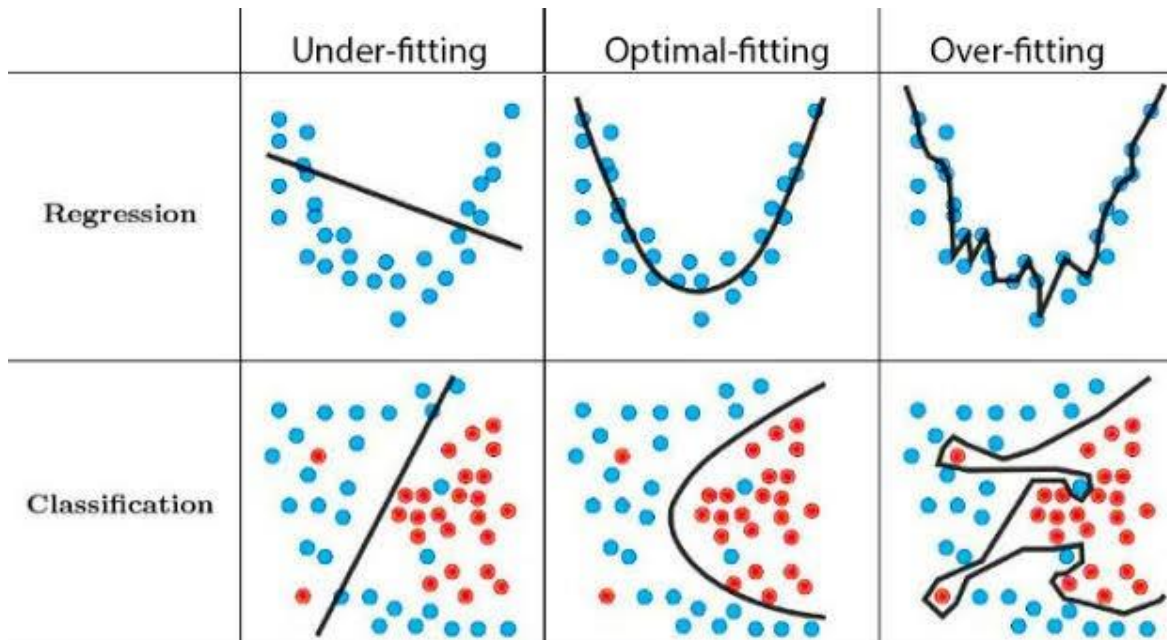
$$y = g(g((x_1 w_1 + x_2 w_2 + \theta_1)) w_5 + g((x_1 w_3 + x_2 w_4 + \theta_1)) w_6 + \theta_2)$$

$$\text{Erro} = \frac{1}{2} (y - \text{real})^2$$

$$\frac{\partial \text{Erro}}{\partial w_1} = \frac{\partial \text{Erro}}{\partial y} * \frac{\partial y}{\partial h_1} * \frac{\partial h_1}{\partial w_1}$$



Problemas no treinamento



Modelo não captura a
tendência dos dados

Modelo captura o
ruído dos dados



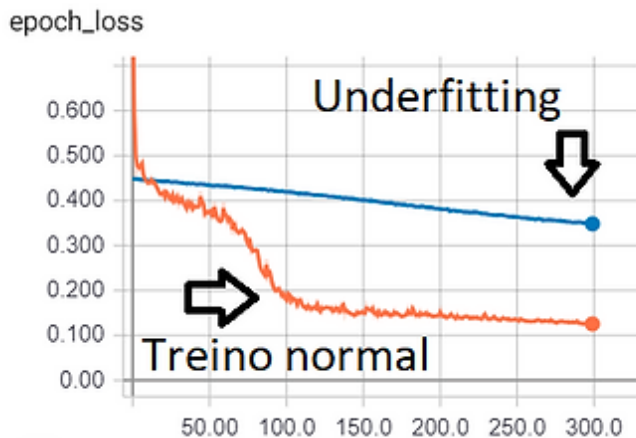
Problemas no treinamento

- Underfitting visto no gráfico de função de perda por época de treinamento



Problemas no treinamento

- Underfitting visto no gráfico de função de perda por época de treinamento





Problemas no treinamento

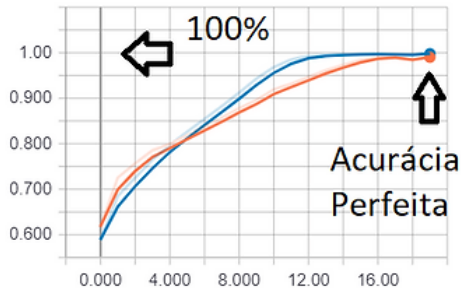
- Overfitting visto no gráfico de função de perda por época de treinamento



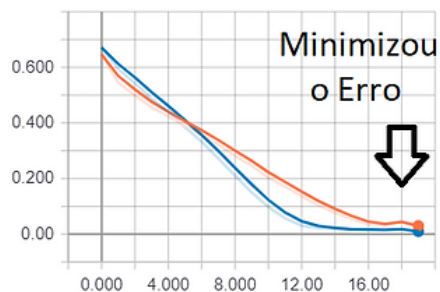
Problemas no treinamento

- Overfitting visto no gráfico de função de perda por época de treinamento

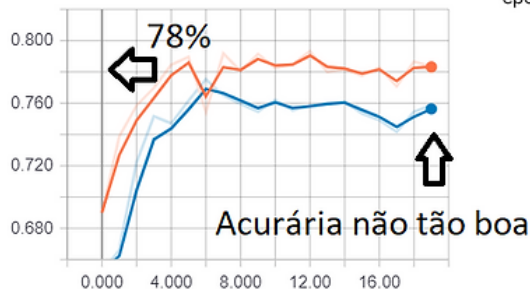
epoch_acc TREINAMENTO



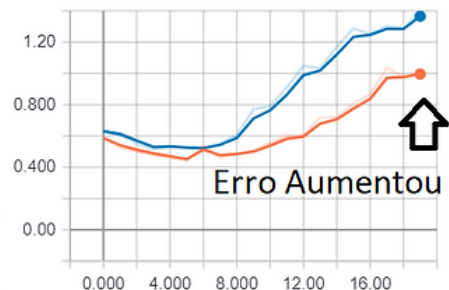
epoch_loss



epoch_val_acc VALIDAÇÃO



epoch_val_loss



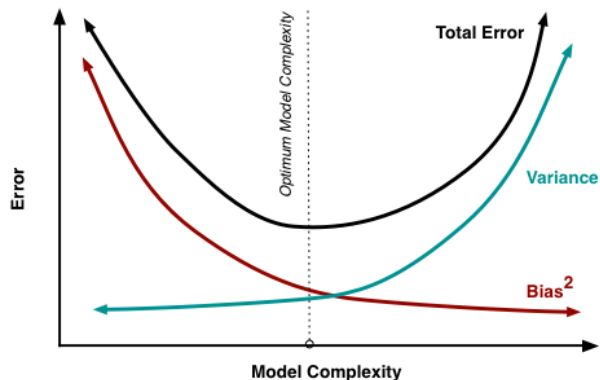
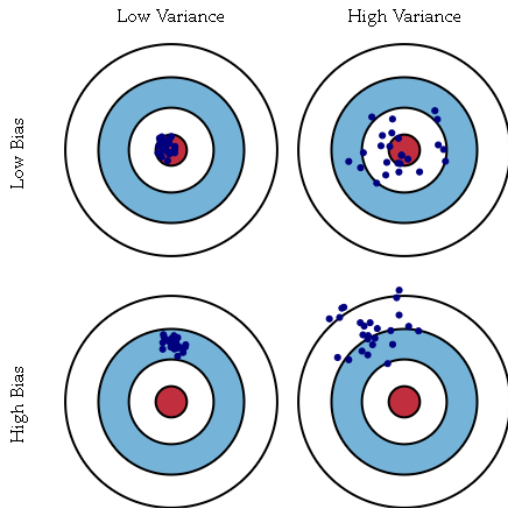


Problemas no treinamento

Performance no conjunto de	Underfitting	Overfitting
Treinamento	Ruim	Bom
Teste	Ruim	Ruim



Problemas no treinamento



- Bias tem a ver com as simplificações feitas pelo modelo para facilitar o aprendizado, a tendência do modelo em prever erroneamente.

- Variância se refere a sensibilidade do modelo às variações nos dados, quão espalhadas estão as respostas



Obrigado!

Perguntas?

🕒 krigor.silva@edu.udesc.br