

CIÊNCIA DE DADOS APLICADA A ANÁLISE ESPORTIVA UTILIZANDO PYTHON AVANÇADO

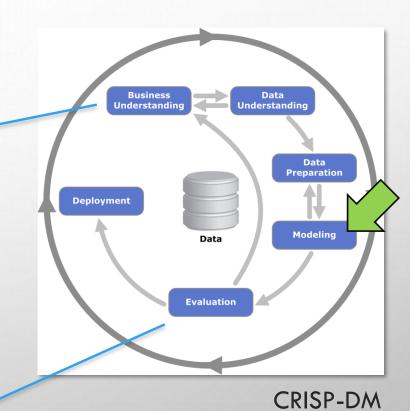
MACHINE LEARNING: CLASSIFICAÇÃO

DIEGO RODRIGUES DSC

INFNET

CRONOGRAMA

ÚMERO	ÁREA	AULA	TRABALHOS
1	Intro	Introdução a Disciplina e Organização do Ambiente	
2	Dados	Coleta de Dados e Sensoriamento	
3		Variáveis Aleatórias	Grupos
4		Análise Exploratória	
5		Estatísticas para Ranqueamento	
6	Estatística	Ranqueamento Estatístico : ELO	
7		Ranqueamento Estatístico : Glicko	
8		Ranqueamento Estatístico : TrueSkill	
9		Ranqueamento Estatístico : XELO	Base de Dados
10		Modelos de Aprendizado de Máquina	
11		Machine Learning: Classificação	
12	ML	Machine Learning: Regressão	
13		Machine Learning: Agrupamento	Pesquisa
14		Machine Learning: Visão Computacional	
15		Aplicações & Artigos: Esportes Independentes	Modelo
16	Esportes	Aplicações & Artigos: Esportes de Objeto	
17	Laportes	Aplicações & Artigos: Esportes de Combate	
18		Aplicações & Artigos : Betting	
19		Workshop	



AGENDA

- PARTE 1 : TEORIA
 - CONCEITOS
 - REDES NEURAIS
 - TREINANDO UMA REDE NEURAL
 - CASE : CLASSIFICAÇÃO DE JOGADAS DE FUTEBOL AMERICANO

SETUP INICIAL DO AMBIENTE PYTHON



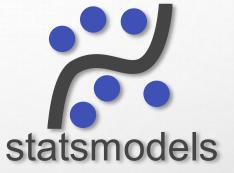
4. Variáveis Aleatórias



5. Visualização

6. Estimação e





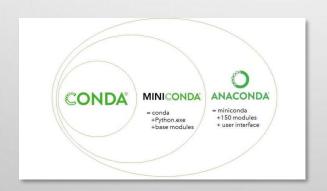


7. Machine Learning





1. Editor de Código



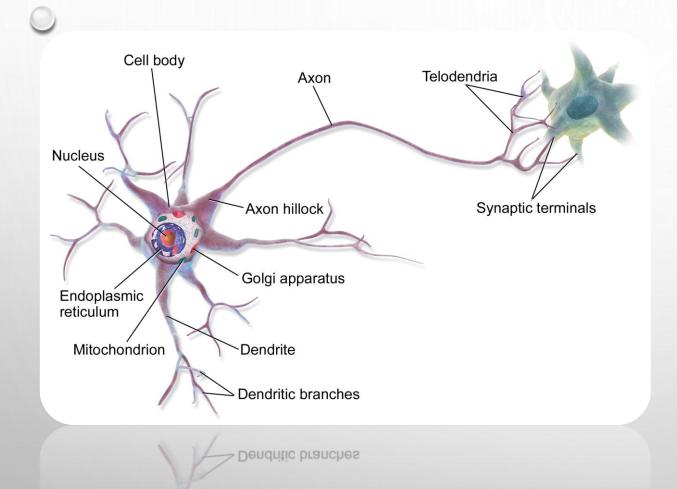
2. Gestor de Ambiente



3. Ambiente
Python do Projeto



3. Notebook Dinâmico



INSPIRAÇÃO BIOLÓGICA



HISTÓRIAS DAS REDES NEURAIS

Perceptron de Rosenblatt 1954

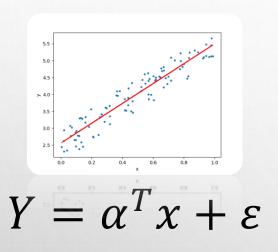
Backpropagation Rumelhart, Hinton & Williams 1986

LSTM ~ 2003

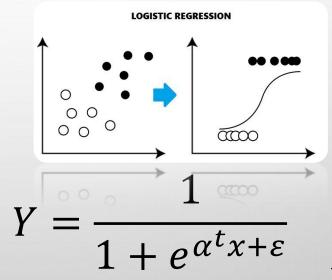
Tensorflow ~ 2015

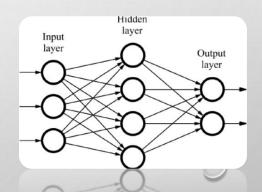
O APROXIMADOR UNIVERSAL

$$Y = F(X) + \varepsilon$$

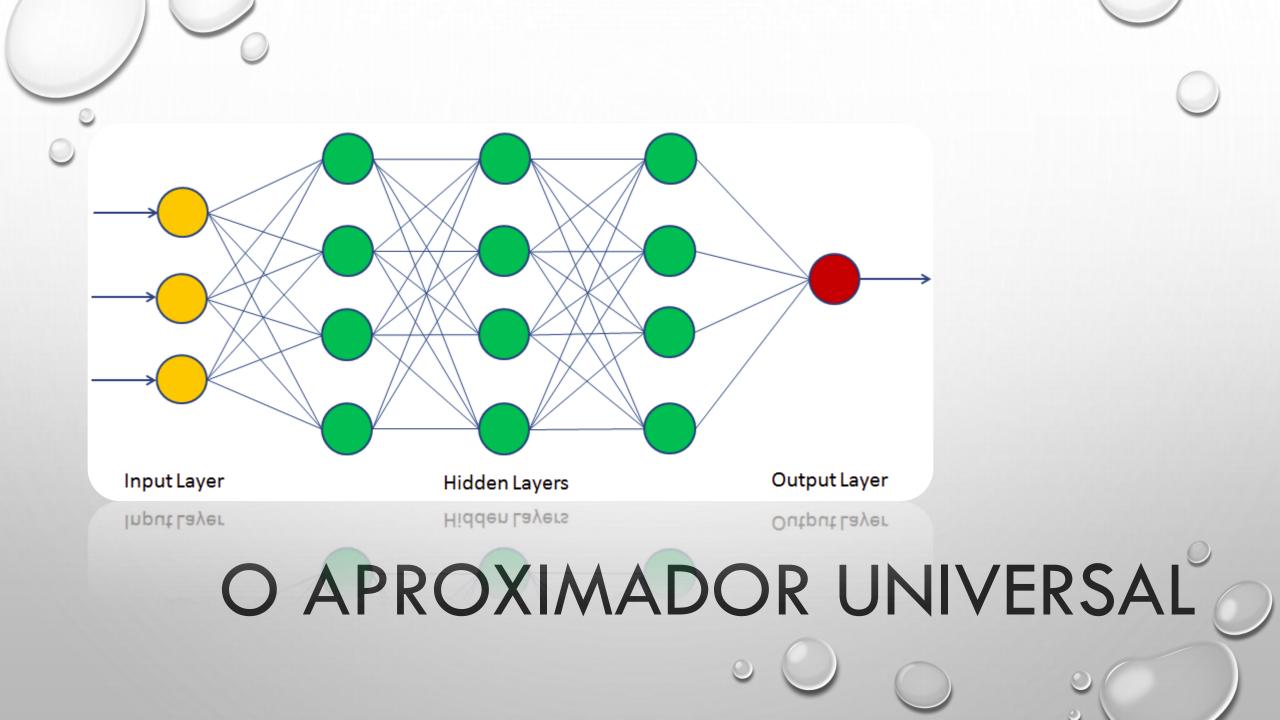


$$Y = X\alpha + \varepsilon$$





$$Y = \varphi(x) + \varepsilon$$



Activation function	Equation	Example	1D Graph
Unit step (Heaviside)	$\phi(z) = \begin{cases} 0, & z < 0, \\ 0.5, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Sign (Signum)	$\phi(z) = \begin{cases} -1, & z < 0, \\ 0, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Linear	$\phi(z) = z$	Adaline, linear regression	
Piece-wise linear	$\phi(z) = \begin{cases} 1, & z \ge \frac{1}{2}, \\ z + \frac{1}{2}, & -\frac{1}{2} < z < \frac{1}{2}, \\ 0, & z \le -\frac{1}{2}, \end{cases}$	Support vector machine	
Logistic (sigmoid)	$\phi(z) = \frac{1}{1 + e^{-z}}$	Logistic regression, Multi-layer NN	
Hyperbolic tangent	$\phi(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$	Multi-layer Neural Networks	
Rectifier, ReLU (Rectified Linear Unit)	$\phi(z) = \max(0, z)$	Multi-layer Neural Networks	
Rectifier, softplus Copyright © Sebastian Raschka 2016 (http://sebastianraschka.com)	$\phi(z) = \ln(1 + e^z)$	Multi-layer Neural Networks	
Rectifier, softplus Copyright © Sebastian Raschka 2016 (http://sebastianraschka.com)	$\phi(z) = \ln(1 + e^z)$	Multi-layer Neural Networks	-
Unit)		Networks	

FUNÇÕES DE ATIVAÇÃO



GOOGLE TENSORFLOW PLAYGROUND



TREINANDO UMA REDE NEURAL



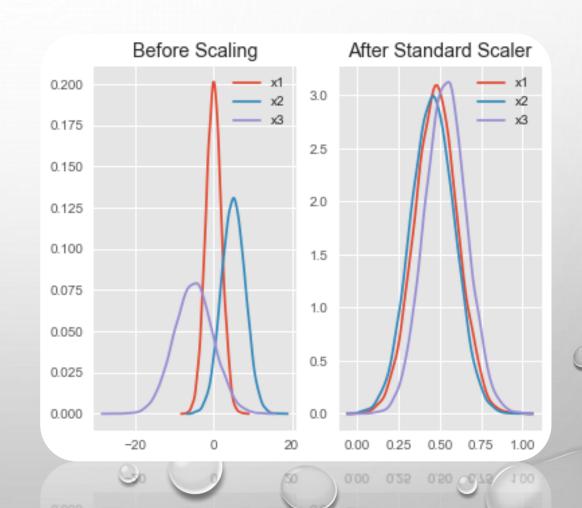
REPRESENTAÇÃO

- Normalização
 - · Garantir que as variáveis independentes possuem a mesma escala
- Análise de Componentes Principais
 - Garantir que as variáveis independentes sejam descorrelacionadas



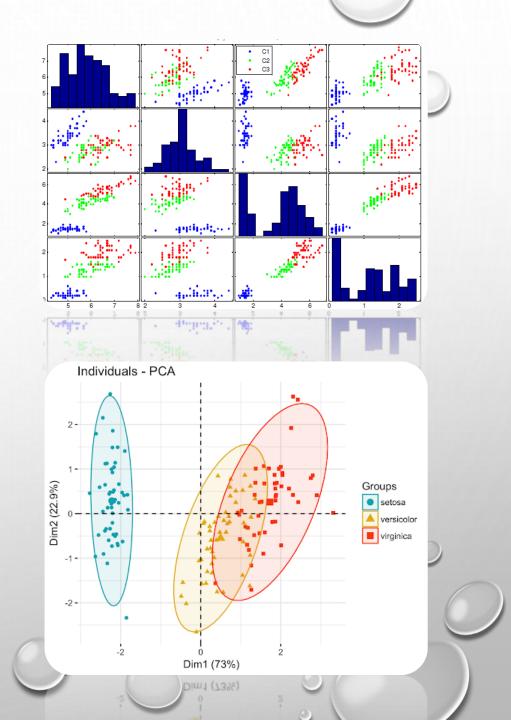
NORMALIZAÇÃO

- Garantir que as variáveis independentes possuem a mesma escala
- Mesmo efeito numérico na otimização independente da escala.



ANÁLISE DE COMPONENTES PRINCIPAIS

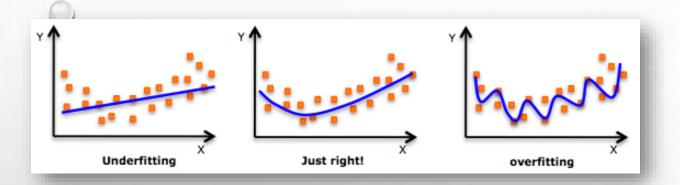
- Garantir que as variáveis independentes sejam descorrelacionadas.
- Identificar novas direções com maior concentração de energia / informação.

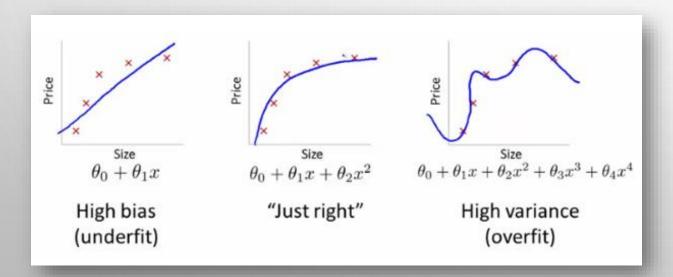


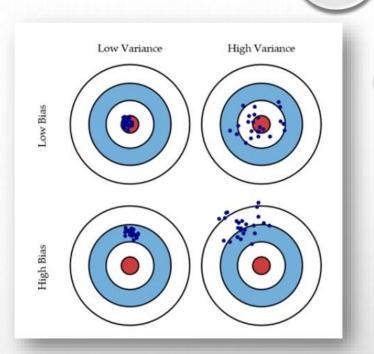


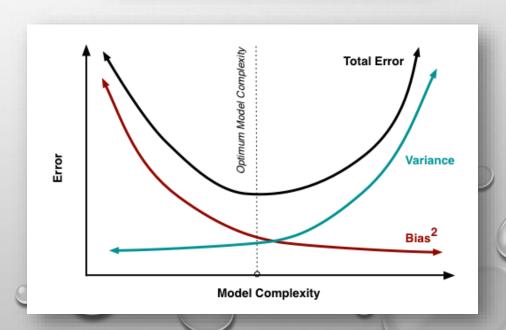
MODELING

BIAS x VARIANCE









REGULARIZAÇÃO

In mathematics, statistics, finance,^[1] and computer science, particularly in machine learning and inverse problems, **regularization** is a process that changes the result answer to be "simpler". It is often used to obtain results for ill-posed problems or to prevent overfitting.^[2]

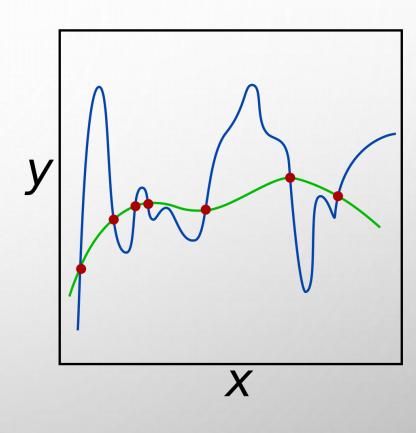
Although regularization procedures can be divided in many ways, the following delineation is particularly helpful:

- Explicit regularization is regularization whenever one explicitly adds a term to the optimization
 problem. These terms could be priors, penalties, or constraints. Explicit regularization is commonly
 employed with ill-posed optimization problems. The regularization term, or penalty, imposes a cost on
 the optimization function to make the optimal solution unique.
- Implicit regularization is all other forms of regularization. This includes, for example, early stopping, using a robust loss function, and discarding outliers. Implicit regularization is essentially ubiquitous in modern machine learning approaches, including stochastic gradient descent for training deep neural networks, and ensemble methods (such as random forests and gradient boosted trees).

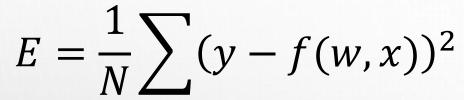
In explicit regularization, independent of the problem or model, there is always a data term, that corresponds to a likelihood of the measurement and a regularization term that corresponds to a prior. By combining both using Bayesian statistics, one can compute a posterior, that includes both information sources and therefore stabilizes the estimation process. By trading off both objectives, one chooses to be more addictive to the data or to enforce generalization (to prevent overfitting). There is a whole research branch dealing with all possible regularizations. In practice, one usually tries a specific regularization and then figures out the probability density that corresponds to that regularization to justify the choice. It can also be physically motivated by common sense or intuition.

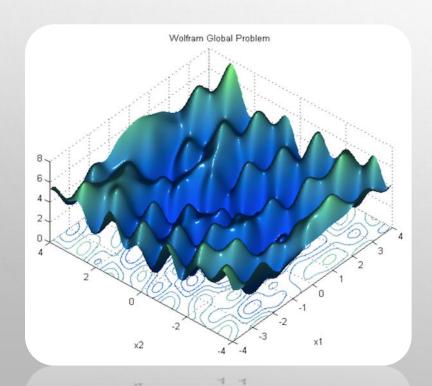
In machine learning, the data term corresponds to the training data and the regularization is either the choice of the model or modifications to the algorithm. It is always intended to reduce the generalization error, i.e. the error score with the trained model on the evaluation set and not the training data.^[3]

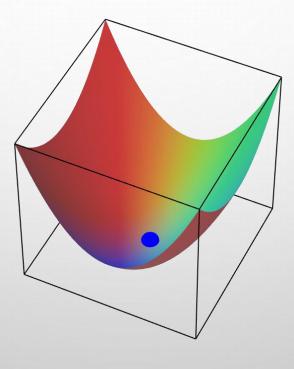
One of the earliest uses of regularization is Tikhonov regularization (ridge regression), related to the method of least squares.

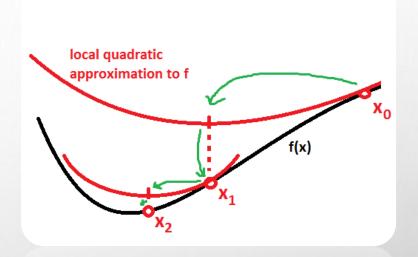


SUPERFÍCIE DO ERRO MÉDIO QUADRÁTICO



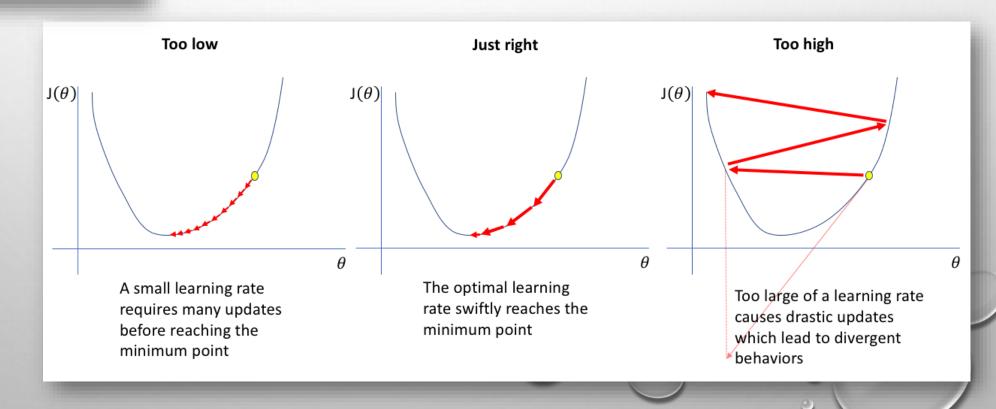




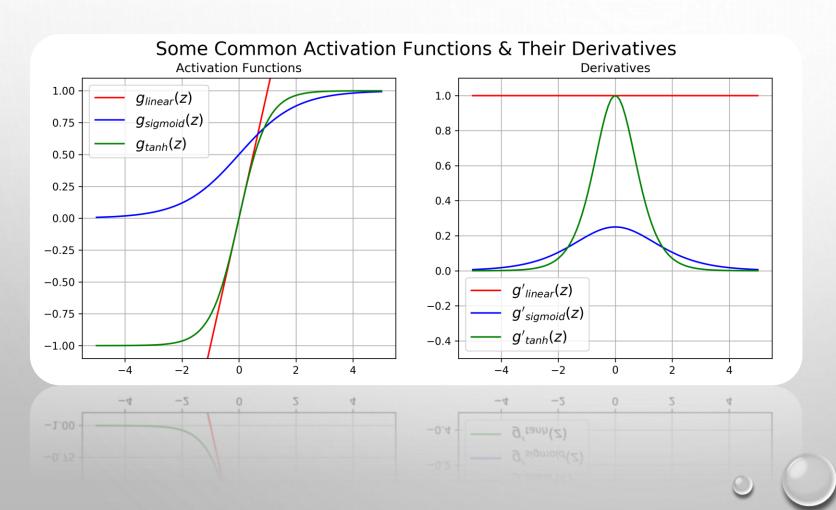


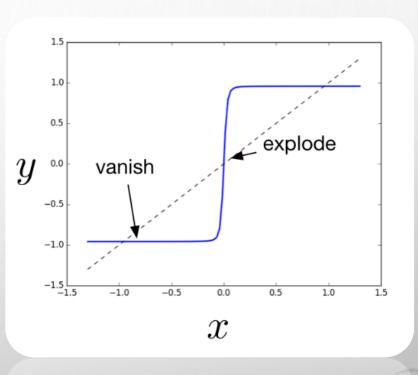
$$\Delta w_{ij} = (\eta * \frac{\partial E}{\partial w_{ij}})$$
weight learning weight increment rate gradient
$$\Delta w_{ij} = (\eta * \frac{\partial E}{\partial w_{ij}}) + (\gamma * \Delta w_{ij}^{t-1})$$
momentum weight increment, previous iteration

ALGORITMO DO GRADIENTE DESCENDENTE



O PROBLEMA DA DISSIPAÇÃO DO GRADIENTE





OTIMIZADORES (REGULARIZADOS)

$$v_{t}^{w} = v_{t-1}^{w} + (\nabla w_{t})^{2}$$

$$v_{t}^{w} = \beta * v_{t-1}^{w} + (1 - \beta)(\nabla w_{t})^{2}$$

$$w_{t+1} = w_{t} - \frac{\eta}{\sqrt{v_{t}^{w} + \epsilon}} * \nabla w_{t}$$

$$w_{t+1} = w_{t} - \frac{\eta}{\sqrt{v_{t}^{w} + \epsilon}} * \nabla w_{t}$$

$$v_t^b = v_{t-1}^b + (\nabla b_t)^2$$

$$b_{t+1} = b_t - \frac{\eta}{\sqrt{v_t^b + \epsilon}} * \nabla b_t$$

$$\Delta b_t = 0$$

$$\Delta b_t + \epsilon$$

$$\Delta b_t = 0$$

$$\Delta b_t = 0$$

$$v_t^w = \beta * v_{t-1}^w + (1 - \beta)(\nabla w_t)^2$$
$$w_{t+1} = w_t - \frac{\eta}{\sqrt{v_t^w + \epsilon}} * \nabla w_t$$

$$v_t^b = \beta * v_{t-1}^b + (1 - \beta)(\nabla b_t)^2$$

$$b_{t+1} = b_t - \frac{\eta}{\sqrt{v_t^b + \epsilon}} * \nabla b_t$$

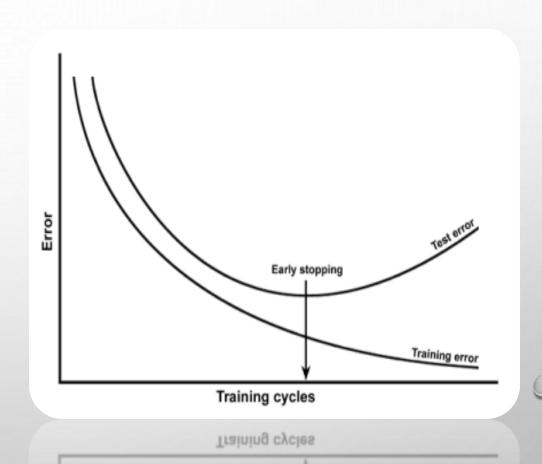
$$v_t^b + \epsilon$$

$$m_{t} = \beta_{1} * m_{t-1} + (1 - \beta_{1}) * \nabla w_{t}$$
 $v_{t} = \beta_{2} * v_{t-1} + (1 - \beta_{2}) * (\nabla w_{t})^{2}$
 $\hat{m}_{t} = \frac{m_{t}}{1 - \beta_{1}^{t}} \qquad \hat{v}_{t} = \frac{v_{t}}{1 - \beta_{2}^{t}}$
 $w_{t+1} = w_{t} - \frac{\eta}{\sqrt{\hat{v}_{t} + \epsilon}} * \hat{m}_{t}$

ADAM

PARADA PREMATURA DO TREINAMENTO

- Aumento no Erro de Validação (Teste)
- Estabilidade da Figura de Mérito no Treino



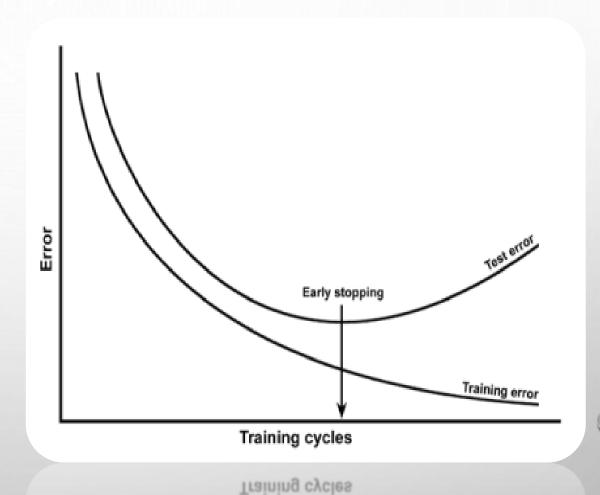


VALIDATION



CRITÉRIOS DE PARADA

- Aumento no Erro de Validação
- Estabilidade da Figura de Mérito no Treino
- Estabilidade de Figura de mérito no Teste

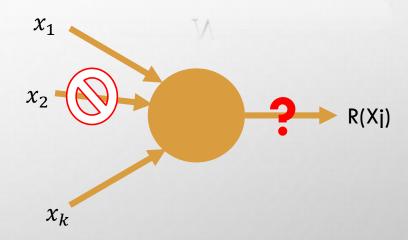




RELEVÂNCIA

- ANÁLISE NÃO-LINEAR DA IMPORTÂNCIA DOS ATRIBUTOS.
- ELIMINA-SE UM ATRIBUTO DA ENTRADA DA REDE, MEDINDO O EFEITO NA CAPACIDADE PREDITIVA.
- A COMPARAÇÃO DAS RELEVÂNCIAS NORMALIZADAS PERMITE IDENTIFICAR OS ATRIBUTOS MAIS IMPORTANTES.

$$R(X_j) = \frac{\sum_{i=1}^{N} \left| |\hat{y}(\mathbf{x_i}) - \hat{y}(\mathbf{x_i}|_{x_{ij} = \bar{x}_j}) \right| |^2}{N}$$



A redução na quantidade de atributos permite simplificar o modelo, salientando os atributos direcionadores da classe.



CLASSIFICAÇÃO DE JOGADAS DE FUTEBOL AMERICANO



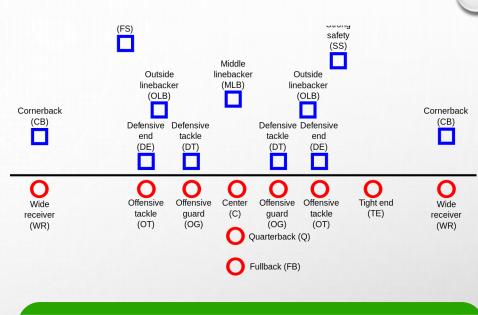
FUTEBOL AMERICANO

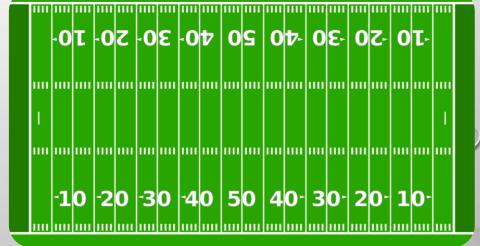
Campo com 54x120 jardas.

Partida disputada em 2 tempos.

Cada tempo é dividido em dois quartos de 15 minutos.

Objetivo: conquistar território e chegar a endzone adversária, para marcar um touchdown ou field goal.





ANÁLISE DE ATRIBUTOS DIRECIONADORES

Representação

- Dados da temporada de 2016 do Indianapolis Colts.
- Quarterback Payton Manning
- Atributos extraídos do Play-by-Play.

Treinamento

- Ensemble de Redes Neurais Multicamadas.
- Treinamento Jacknife.
- Análise de Relevância
- Figura de Mérito: Acurácia



ATRIBUTOS COLETADOS DO PLAY-BY-PLAY

N	Atributo	Tipo
1	Distância para o First Down	Contínuo
2	Posição no Campo	Contínuo
3	Diferença no Placar	Contínuo
4	Quarto de Jogo	Discreto
5	Tempo	Contínuo
6	Dois Minutos Finais	Binário
7	Última Jogada - Passe	Binário
8	Última Jogada - Corrida	Binário
9	Última Jogada - Times Especiais	Binário
10	Última Jogada - Jardas	Contínuo
11	Tipo de Joagada (Corrida/Passe)	Classe

11	Tipo de Joagada (Corrida/Passe)	Classe
10	Última Jogada - Jardas	Contínuo
	Ultima Jogada - Times Espe⊕is	(Bi) SIIO



- REDE NEURAL DO TIPO MULTILAYER PERCEPTRON (MLP) COM UMA CAMADA ESCONDIDA.
 - UMA REDE FOI TREINADA PARA OS DADOS DE CADA DOWN (COMITÊ).
- TREINAMENTO USANDO O ALGORITMO RPROP.
- TREINAMENTO JACKNIFE USANDO ACURÁCIA COMO FIGURA DE MÉRITO – VALIDAÇÃO CRUZADA COM 5 PARTIÇÕES.
 - FASE 1 EXTRAÇÃO DOS DIRECIONADORES.
 - FASE 2 VALIDAÇÃO.



RESULTADOS

REDES NEURAIS: TREINAMENTO FASE 1

Tentativa	Passe	Corrida
Primeiro down	113	135
Segundo down	100	86
Terceiro down	85	12
Total	298	233

Total	298	233

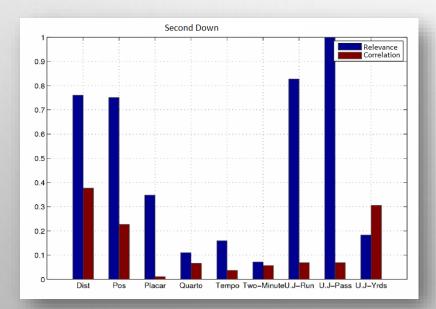
Tentativa	NE	μ (%)	$\sigma(\%)$	Melhor(%)
10 down	20	61	6	71
20 down	9	67	6	78
3o down	8	94	5	97



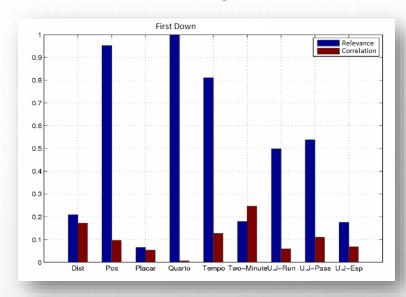
RESULTADOS DA RELEVÂNCIA

Atributo	1D	2D	3D
Distância	21	76	52
Posição	95	75	98
Placar	6	34	41
Quarto	100	11	100
Tempo	81	15	63
2 Minutos	17	7	17
U.J. Corrida	49	82	4
U.J. Passe	53	100	31
U.J. Especial	17	-	-
U.J. Jardas	-	18	10

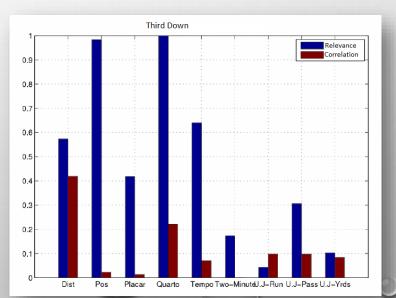
U.J. Passe, U.J. Corrida, Distância



Quarto, Posição, Tempo



Quarto, Posição, Tempo, Distância

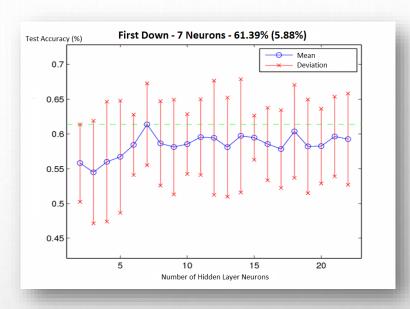


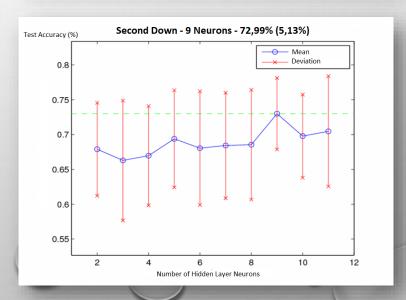
TREINAMENTO 2 – RESULTADO FINAL

Modelo	NE	μ (%)	$\sigma(\%)$	Melhor (%)
NN-1D	7	61	5	74
NN-2D	9	73	5	81
NN-3D	8	94	5	97
NN	10	73	-	73
See5	-	69	-	69
ID3	-	64	-	64

5/10 atributos selecionados como direcionadores para o primeiro down e segundo down.

Eficiência dos direcionadores selecionados confirmado por 80% de acurácia.





DESAFIO: ANÁLISE EXPLORATÓRIA DOS DADOS DO P. MANNING

PRÓXIMA AULA LEITURA: REGRESSÃO