

Trabalho de Machine Learning

João Victor Girard

Data Set utilizado

Foi escolhido o data set:

<http://archive.ics.uci.edu/ml/datasets/Car+Evaluation>

Car Evaluation Data Set

Download: [Data Folder](#), [Data Set Description](#)

Abstract: Derived from simple hierarchical decision model, this database may be useful for testing constructive induction and structure discovery methods.



Data Set Characteristics:	Multivariate	Number of Instances:	1728	Area:	N/A
Attribute Characteristics:	Categorical	Number of Attributes:	6	Date Donated	1997-06-01
Associated Tasks:	Classification	Missing Values?	No	Number of Web Hits:	882092

Data Set

Neste data set existem os seguintes elementos:

Classes:

- **unacc, acc, good, vgood**

Atributos:

- **buying:** vhigh, high, med, low.
- **maint:** vhigh, high, med, low.
- **doors:** 2, 3, 4, 5more.
- **persons:** 2, 4, more.
- **lug_boot:** small, med, big.
- **safety:** low, med, high.

Probabilístico

É lido cada um dos exemplos do batch de treinamento e somado o número de vezes que o evento ocorreu. Ao final da contagem, é iniciado a avaliação dos objetos no batch de testes.

Probabilístico

Esta avaliação é feita de modo que cada evento é tomado como um evento independente e a probabilidade de cada um dos eventos lidos, dada a quantidade de elementos total, é multiplicada uma a uma para ser determinada a probabilidade da classe ocupada.

Possuindo as probabilidades de cada classe para um dado evento, a classe de maior probabilidade é selecionada.

Probabilístico

Tempo de execução total: $< 1s$

Feito isso, a maior acurácia desse modelo encontrada foi:

>>>> Accuracy is: **0.751 of 1382 subjects for training (80.0%) and 345 subjects for testing**

Probabilístico

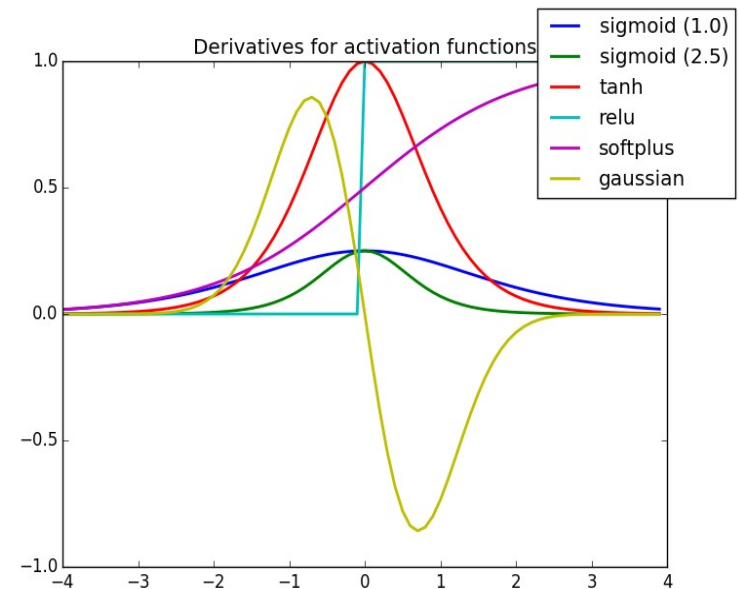
Ponto negativo detectado:

Dataset utilizado tem uma probabilidade muito mais alta de ocorrer uma classe do que as outras 3. Isso faz com que este método encontre sempre os dados para esta classe destoante e não identifique para as outras, gerando assim perdas grandes na acurácia.

Rede Neural

Neste modelo foi utilizado uma rede neural codificada através do **TensorFlow**. A função de ativação utilizada foi a Relu e a taxa de aprendizado foi de 0.07. Foi utilizado somente um neurônio e foram realizadas 2000 épocas.

Tempo de execução total: +- **9s**



Rede Neural

O erro na época 0 é 1.2451559

O erro na época 500 é 0.8332894

O erro na época 1000 é 0.82573855

O erro na época 1500 é 0.8225612

O erro na época 1900 é 0.8205697

A acurácia é: 0.7008102

A Classe de um carro $[[2, 3, 1, 2, 0, 0, 0]]$ é: $[0]$

A classe do elemento é o último atributo do mesmo

Rede Neural

Ponto negativo detectado:

Não consegui implementar uma função funcional que detectasse bem as 4 classes disponíveis, então a utilizada foi uma com o aspecto **One Hot**, detectando assim a classe que possui o maior número de elementos do batch de treinamento.

OBS: Variações em ambos os teste de parâmetros não foram aqui detalhadas, pois as mesas apresentavam o mesmo resultado dado os pontos negativos de cada um dos métodos.

Conclusão

	Tempo	Accu
Probabilístico	< 1 s	0.752
Rede Neural	+ - 9 s	0.701