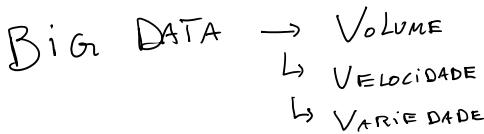
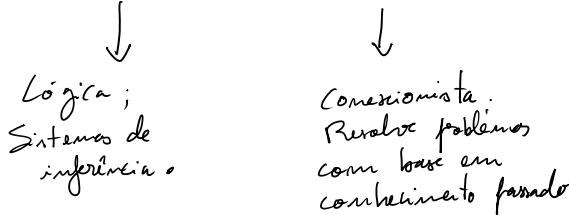


Hard Computing vs Soft Computing

(Computação simbólica vs Computação não simbólica)

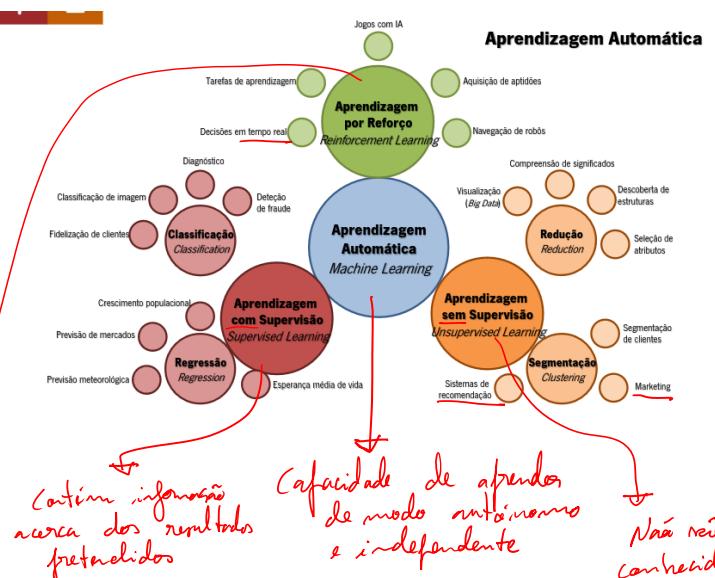


- Machine learning refers to a system capable of the autonomous acquisition and integration of knowledge. This capacity to learn from experience, analytical observation, and other means, results in a system that can continuously self-improve and thereby offer increased efficiency and effectiveness.

<http://www.aaai.org/AITopics/html/machine.html>

- Machine learning refers to a system capable of the autonomous acquisition and integration of knowledge. This capacity to learn from experience, analytical observation, and other means, results in a system that can continuously self-improve and thereby offer increased efficiency and effectiveness.

<http://www.aaai.org/AITopics/html/machine.html>



→ É possível efectuar
uma avaliação sobre
se os resultados
produzidos são bons
ou maus.

Reconfirma / Paralisa

" Sabem-se os meios
dos problemas
que se quer "

DATA - DRIVEN
ALGORITHMS

- ↓
- Segmentação
 - Associação

METODOLOGIAS DE ANÁLISE DE DADOS

• CRISP-DM (1)

Cross Industry Standard
Process for Data Mining

• SEMMA (2)

Sample, Explore, Modify, Model
and Assess

OBJETIVOS

(1)

- Definir um processo de Análise de Dados para a indústria;
- Construir e disponibilizar ferramentas de apoio;
- Assegurar a qualidade dos projetos de Análise de Dados;
- Reduzir os conhecimentos específicos necessários para conduzir um processo de Análise de Dados.

6 etapas

- Estudo do negócio → compreender objetivos
- Estudo dos dados → obter dados
- Preparação dos dados → seleção + limpeza
- Modelação → Experimentação
- Avaliação → Conformidade com objetivos
- Desenvolvimento / Deployment
↳ colocação do modelo em produção



20

↳ Não gosto muito.
Demasiado business-related para a ideia que fizemos para o TP.

SEMMA

- Extração de dados do universo do problema;
- Baseia o processo de *Data Mining* no conceito de "amostra" do problema;
- Amostra pequena e significativa;
- Proporciona flexibilidade e rapidez no tratamento dos dados.
 - "Data Mining é o processo de **extrair conhecimento e relações complexas** de grandes volumes de dados."
- Motivação:
 - necessidade de definir, padronizar e integrar sistemas ou processos de *Data Mining* nos ciclos de produção.
- Desenvolvimento focado na ferramenta [SAS Enterprise Miner](#).



• SAMPLE

- Extração de dados do universo do problema;
- Baseia o processo de *Data Mining* no conceito de "amostra" do problema;
- Amostra pequena e significativa;
- Proporciona flexibilidade e rapidez no tratamento dos dados.

• EXPLORE

- Exploração visual e/ou numérica das tendências;
- Refinamento do processo de descoberta (*mining*);
- Técnicas estatísticas: regressão linear, mínimos quadrados, distribuição de Poisson, etc.;
- Procura de tendências imprevistas nos dados;

• MODIFY

- Concentração de todas as modificações necessárias;
- Inclusão de informação;
- Seleção ou introdução de novas variáveis;
- Objetivo: criar, selecionar e adaptar variáveis para a próxima etapa;

• MODELAGÃO

• AVALIAÇÃO

- Aferição do desempenho do modelo construído para *Data Mining*;
- Aplicação do modelo a uma amostra de dados de teste;
- Procedimento de ajuste do modelo.

▪ Fases CRISP-DM:

- Estudo do negócio;
- Estudo dos dados;
- Preparação dos dados;
- Modelação;
- Avaliação;
- Desenvolvimento.



▪ Processo SEMMA:

- Amostragem;
- Exploração;
- Modificação;
- Modelação;
- Avaliação.

PREPARAÇÃO DE DADOS

↳ Transformar os dados para forma a que a informação esteja adequada à ferramenta

- Os dados recolhidos do "mundo real":

- são incompletos;
- contêm lixo;
- podem conter inconsistências.

- Discretização;
(classes etárias)

- Limpeza;
(nº BI)

- Integração;
(fontes)

- Transformação;
(diários/mensais)

- Redução de dados.
(moradas/regiões)

- Os dados recolhidos do "mundo real":

- são incompletos:

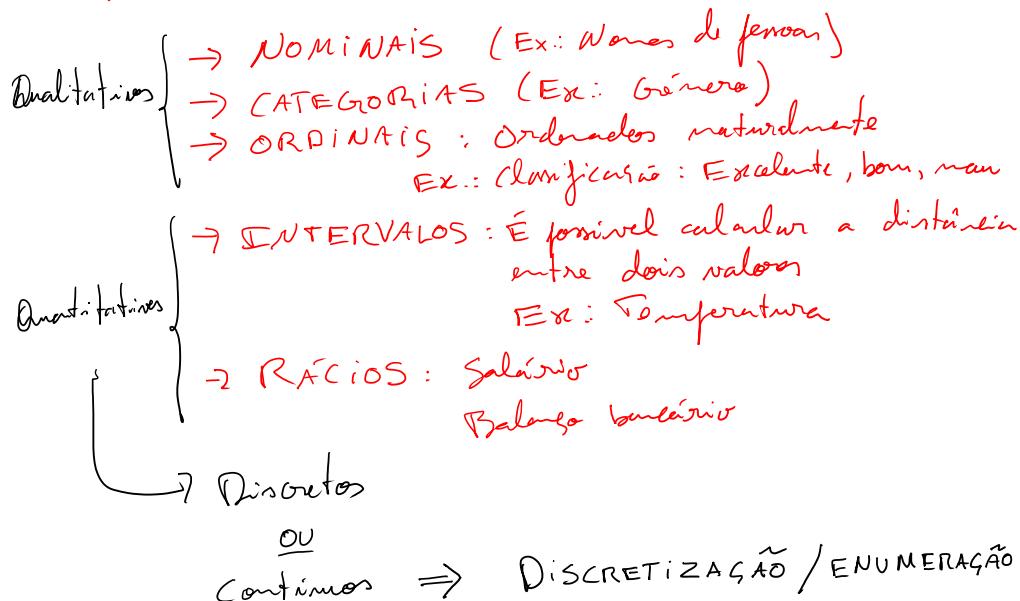
- falta de valores em alguns atributos;
- falta de alguns atributos;
- dados agregados ou generalizados;
- Código postal: 4710-... Braga;
- Nº de filhos: "";

- contêm lixo;

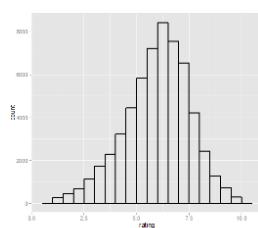
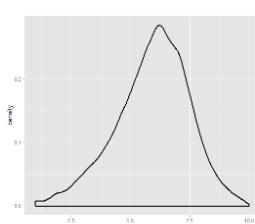
- podem conter inconsistências.

↳ Muitas fontes de dados: BD's, ficheiros, faxel, web, etc.

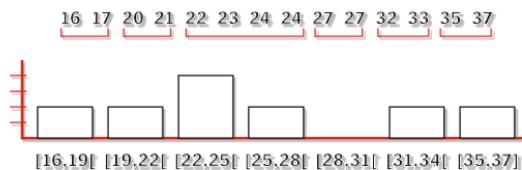
TIPOS DE DADOS



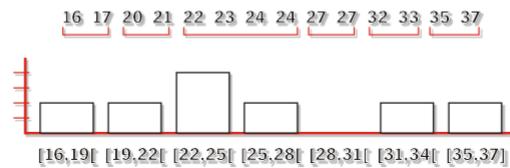
- Os métodos mais utilizados (Naive Bayes, CHAID, etc.), requerem valores discretos;
- Redução do tamanho dos dados;
- Método utilizado para produzir sumariação dos dados;
- (Sinónimo de binning)



- Equal-width binning:
- Divide a gama de valores em N intervalos de igual largura, resultando numa grelha uniforme;
- Sendo A e B os limites da gama de valores, a largura dos intervalos será $L = (B - A) / N$:



- Equal-width binning:
- Divide a gama de valores em N intervalos de igual largura, resultando numa grelha uniforme;
- Sendo A e B os limites da gama de valores, a largura dos intervalos será $L = (B - A) / N$:



- Normalmente preferida à discretização de igual largura, uma vez que permite evitar o “amontoar” de valores;
- Na prática, utiliza-se uma discretização de “quase-igual” altura, garantindo intervalos mais intuitivos;
- Deverá impedir a dispersão de valores frequentes por diferentes intervalos;
- Deverá criar intervalos separados para valores especiais (“0”).

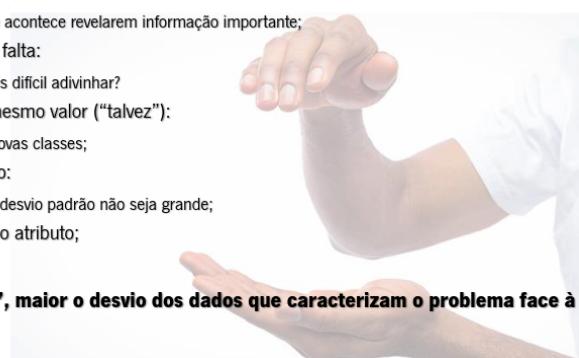
Outros métodos:

- Discretização baseada em entropia
- Discretização baseada em impurezas
- Detecção de limites
- Etc.

- Ignorar os registos onde faltam os dados e lidar, apenas com os dados conhecidos;
 - não aconselhável se a quantidade de dados em falta em cada atributo for elevada;
- Ignorar os atributos onde faltam os dados;
 - não aconselhável se os atributos onde acontece revelarem informação importante;
- Preencher (manualmente) os dados em falta:
 - é mais trabalhoso preencher ou é mais difícil adivinhar?
- Preencher os dados em falta com um mesmo valor (“talvez”):
 - pode criar tendências nos dados ou novas classes;
- Preencher com o valor médio do atributo:
 - pouco impacto negativo, desde que o desvio padrão não seja grande;
- Preencher com o valor mais frequente do atributo;

▪ **Quantos mais valores “inventados”, maior o desvio dos dados que caracterizam o problema face à realidade que o problema ilustra!**

~~EVITAR ADICIONAR DISTORSÃO~~



- Alisamento (*smoothing*);
- Agregação; *Eex.: Sum*
- Generalização; *Hierarquia de conceitos: distrito → cidade → zona*
- Construção de atributos:
 - Construção de novos atributos a partir de outros (cálculo do preço líquido baseado no preço ilíquido e no IVA);
- Uniformização;
- Detecção de valores atípicos.

TRANSFORMAÇÃO
DE DADOS

Normalização: $[0, 1]$
Padronização: Z-score

↳ Por visualização: Bon plot

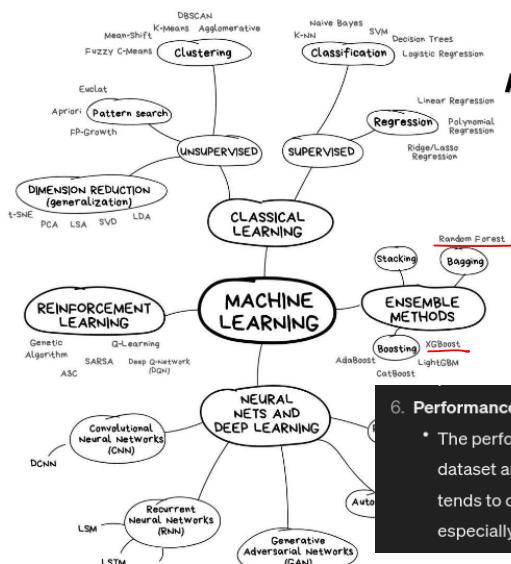


ESTRATÉGIAS DE REDUÇÃO DE DADOS

- Construção de cubos de dados:
 - as operações de agregação são aplicadas de modo a construir cubos de dados;
- Redução de dimensões:
 - remoção de atributos que se mostrem irrelevantes, redundantes ou pouco interessantes para a análise;
 - *Principle Component Analysis (PCA)*;
- Compressão de dados:
 - aplicação de técnicas de compressão ou de transformação para comprimir a representação dos dados originais;
- Redução de quantidade:
 - redução do volume de dados (técnicas paramétricas ou não paramétricas);
- Discretização e generalização de conceitos:
 - redução da quantidade de valores por atributo.



ARVORES DE DECISÃO



AI



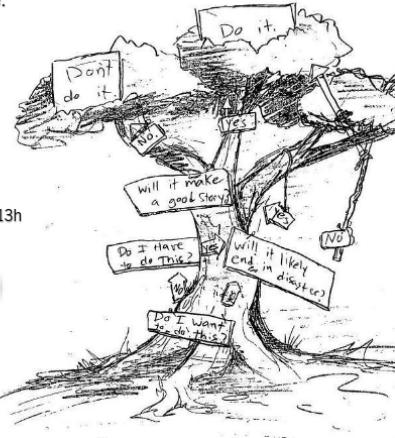
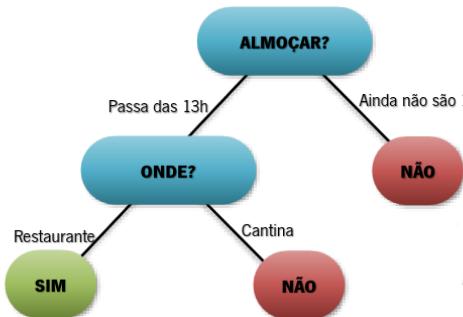
6. Performance:

- The performance of Random Forest and XGBoost depends on the dataset and the specific problem at hand. In general, XGBoost tends to outperform Random Forest on structured/tabular data, especially when there are complex relationships between features.

DECISION TREES

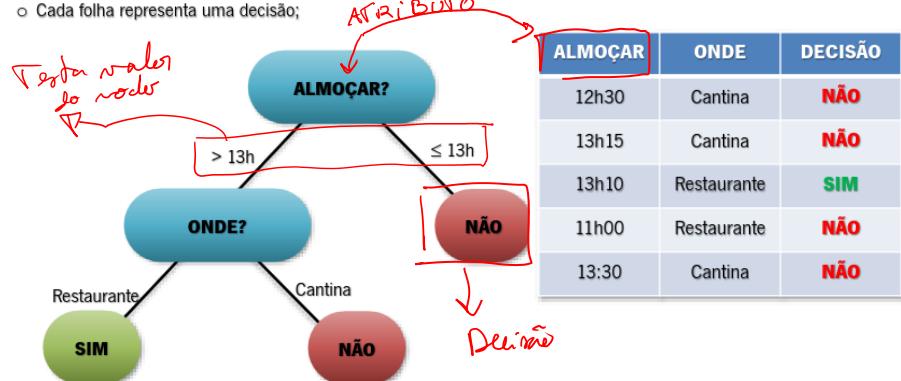
Uma Árvore de Decisão é um grafo hierarquizado (árvore!) em que:

- Cada ramo representa a seleção entre um conjunto de alternativas;
- Cada folha representa uma decisão;



RAMO → Seleção entre conjunto de alternativas
FOLHA → Representa uma decisão

- Cada nodo interno testa um atributo do *dataset*;
- Cada ramo identifica um valor (ou conjunto de valores) do nodo testado;
- Cada folha representa uma decisão;



■ Paradigmas de criação de modelos de decisão:

○ Top-down.

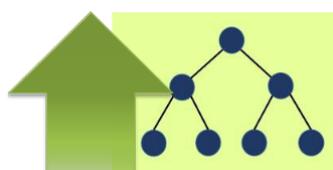
- O modelo é construído a partir do conhecimento de especialistas;
- O “todo” é dividido em “partes”;

○ Bottom-up.

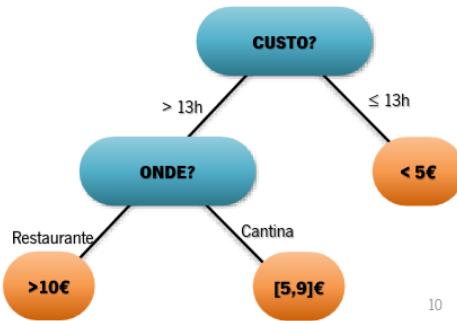
- O modelo é construído pela identificação de relações entre os atributos do *dataset*;
- O modelo é induzido por “generalização” dos dados;

↳ BOTTOM - UP

- Toda a informação sobre cada item de dados (ou objeto) deve estar definido numa coleção fixa e finita de atributos;
- Desta modo, objetos distintos não podem requerer coleções distintas de atributos;
- Quando o conjunto dos níveis de decisão é conhecido *a priori*, a construção do modelo segue um paradigma de aprendizagem supervisionado;
- Quando o conjunto dos níveis de decisão é calculado pelo modelo, a sua construção segue um paradigma de aprendizagem não supervisionado;
- Os níveis de decisão podem ser de 2 tipos:
 - Contínuos: problemas de regressão;
 - Discretos: problemas de classificação;
- Quantidade de objetos >> níveis de decisão;

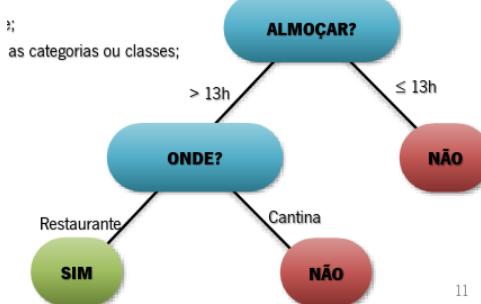


↳ *Continua*:



10

↳ *Discreto*:



11

- Dada uma árvore de decisão (treinada), o processo de decisão desenvolve-se do seguinte modo:

- Começar no nodo correspondente ao atributo "raiz";
- Identificar o valor do atributo;
- Seguir pelo ramo correspondente ao valor identificado;
- Alcançar o nodo relativo ao ramo percorrido;
- Voltar a 2. até que o nodo seja uma "folha";
- O nodo alcançado indica a decisão para o problema.

AL



- Algoritmo ID3: Iterative Dichotomiser 3
link.springer.com/content/pdf/10.1007

- Desenvolvido por Ross Quinlan;
- Constrói uma árvore de decisão, a partir da raiz até às folhas;
- Principal problema:
 - "Qual o melhor atributo para ser a raiz da árvore de decisão?"

- Noção de Entropia:

- Entropia é uma medida da incerteza associada a um conjunto de objetos;
- A entropia identifica o grau de desorganização dos dados;

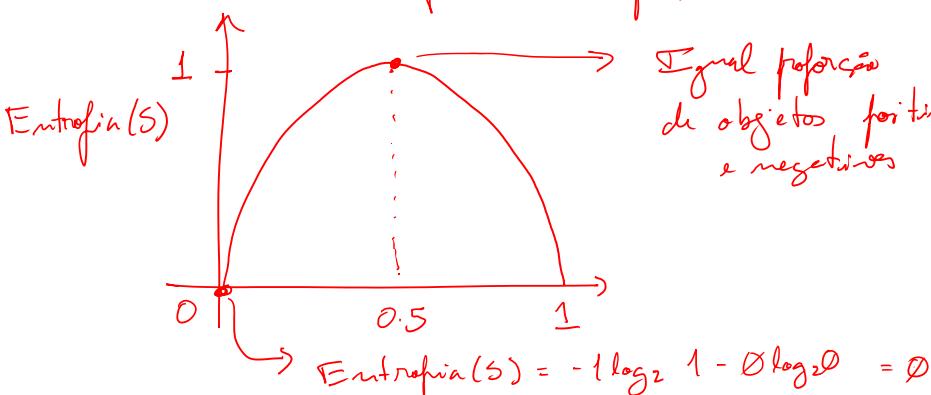


$$\text{Entropia}(S) = -f_+ \log_2 f_+ - f_- \log_2 f_-$$

Exemplo: $f_+ \in [0,1]$

$$f_- \equiv (1-f_+) \in [0,1]$$

Igual proporção de objetos positivos e negativos



Assim, o atributo com a maior redução de entropia é a melhor escolha para ser nodo.

↳ Para reduzir a profundidade da árvore

Ganho de informação:
(Atributo A, Coleção S)

$$\text{Ganho}(S, A) = \text{Entropia}_{\text{original}}(S) - \text{Entropia}_{\text{relativa}}(S)$$

$$= \text{Entropia}(S) - \sum_{v \in \text{valores}(A)} \frac{|S_v|}{|S|} \times \text{Entropia}(S_v)$$

- S cada valor v de todos os valores possíveis do atributo A ;
- S_v subconjunto de S para o qual o atributo A tem o valor v ;
- $|S_v|$ quantidade de objetos em S_v ;
- $|S|$ quantidade de objetos em S ;

3/14

Como calcular o atributo para ser a raiz da árvore de decisão?

$$\circ \text{Entropia}(S) = -\left(\frac{9}{14}\right) \log_2 \left(\frac{9}{14}\right) - \left(\frac{5}{14}\right) \log_2 \left(\frac{5}{14}\right)$$

$$= -(0,643) \times (-0,637) - 0,357 \times (-1,485)$$

$$= 0,410 + 0,530 = 0,940$$

$$\circ \text{Ganho}(S, \text{Vento}) = \text{Entropia}(S) - \left(\frac{8}{14}\right) \text{Entropia}(S_{\text{Fraco}}) - \left(\frac{6}{14}\right) \text{Entropia}(S_{\text{Forte}})$$

$$= 0,940 - \left(\frac{8}{14}\right) \times 0,811 - \left(\frac{6}{14}\right) \times 1,0$$

$$= 0,048$$

$$\circ \text{Entropia}(S_{\text{Fraco}}) = -\left(\frac{6}{8}\right) \log_2 \left(\frac{6}{8}\right) - \left(\frac{2}{8}\right) \log_2 \left(\frac{2}{8}\right) = 0,811$$

$$\circ \text{Entropia}(S_{\text{Forte}}) = -\left(\frac{3}{6}\right) \log_2 \left(\frac{3}{6}\right) - \left(\frac{3}{6}\right) \log_2 \left(\frac{3}{6}\right) = 1,0$$

Dia	Aspetto	Temperatura	Humidade	Vento	Decisão
1	Sol	Calor	Alta	Fraco	Não
2	Sol	Calor	Alta	Forte	Não
3	Nuvens	Calor	Alta	Fraco	Sim
4	Chuva	Amenos	Alta	Fraco	Sim
5	Chuva	Frio	Normal	Fraco	Sim
6	Chuva	Frio	Normal	Forte	Não
7	Nuvens	Frio	Normal	Forte	Sim
8	Sol	Amenos	Alta	Fraco	Não
9	Sol	Frio	Normal	Fraco	Sim
10	Chuva	Amenos	Normal	Fraco	Sim
11	Sol	Amenos	Normal	Forte	Sim
12	Nuvens	Amenos	Alta	Forte	Sim
13	Nuvens	Calor	Normal	Fraco	Sim
14	Chuva	Amenos	Alta	Forte	Não

$$\circ \text{Ganho}(S, \text{Aspecto}) = 0,048$$

$$\circ \text{Ganho}(S, \text{Aspetto}) = 0,246$$

$$\circ \text{Ganho}(S, \text{Temperatura}) = 0,029$$

$$\circ \text{Ganho}(S, \text{Humidade}) = 0,151$$

Atributo com maior ganho de informação

Selecionado para raiz da árvore



Algoritmo C4.5

- ↳ Manipula atributos contínuos e discritos
- Lida com missing values : Threshold
 - ↳ Armazena os missing values que vão serem usados nos cálculos de ganho e entropia
- Permite a atribuição de pesos a atributos
- Permite fazer a PODAR da árvore

■ Porquê?

- Porque uma Árvore de Decisão pode resultar num modelo de decisão "demasiado" adaptado aos dados de treino;
- Cada folha pode representar um caso ou conjunto de casos muito específicos;

outros

■ Algoritmo CHAID: Chi-square Automatic Interaction Detection

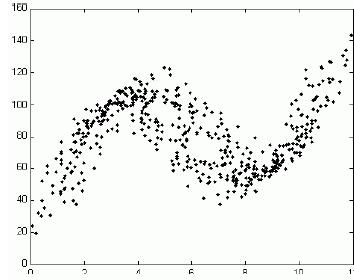
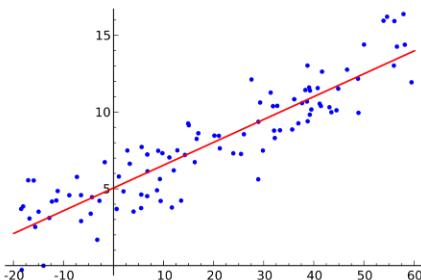
- Opera a separação dos dados em modo multi-nível, enquanto que o CART usa modos binários para essa divisão;
- Adequado para grandes datasets;
- Frequentemente utilizado em estudos de marketing para segmentação de mercados;

■ Pontos fracos:

- Inadequadas para problemas caracterizados por muitas interações entre os atributos;
- Falta de poder expressivo;
- Não isenta a réplicas de subárvore;

TÉCNICAS DE REGRESSÃO

- Quão bem uma determinada variável independente prevê outra variável dependente?
- A regressão é um procedimento estatístico que determina a equação para a reta/curva que melhor se ajusta a um conjunto específico de dados.

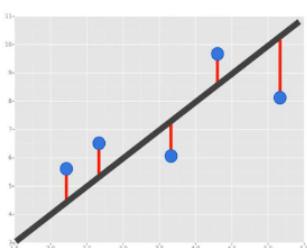


↳ LINEAR

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$$

foi funcional?

- O método dos mínimos quadrados minimiza a soma dos erros ao quadrado:
 - y_i : valor verdadeiro
 - $f(x_i, \beta)$: valor previsto / linha ajustada
- O resíduo para uma observação é a diferença entre a observação (valor y) e a linha ajustada:
 - $r_i = y_i - f(x_i, \beta)$
- O método dos mínimos quadrados procura os parâmetros ótimos, minimizando a soma S:
 - $S = \sum_{i=1}^n r_i^2$



$$0.1^2 = 0.01$$

$2^2 = 4$ → alta representabilidade na média

• Regressão linear múltipla

- A regressão múltipla é usada para determinar o efeito de diversas variáveis independentes, x_1, x_2, x_3, \dots numa variável dependente, y ;
- As diferentes variáveis x_i são combinadas de forma linear e cada uma tem seu próprio coeficiente de regressão:

$$y = a_1 \cdot x_1 + a_2 \cdot x_2 + \dots + a_n \cdot x_n + b + \varepsilon$$

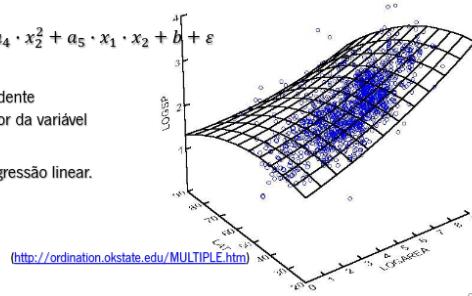
Diversas variáveis independentes, x_i , numa dependente y .

• Regressão polinomial múltipla

- A regressão polinomial múltipla é usada para determinar o efeito de diversas variáveis independentes, x_1, x_2, x_3, \dots numa variável dependente, y ;
- As diferentes variáveis x_1 e x_2 são combinadas de forma polinomial e cada uma tem seu coeficiente de regressão:

$$y = a_1 \cdot x_1 + a_2 \cdot x_1^2 + a_3 \cdot x_2 + a_4 \cdot x_2^2 + a_5 \cdot x_1 \cdot x_2 + b + \varepsilon$$

- Os parâmetros a_i refletem a contribuição independente de cada variável independente x_1 e x_2 , para o valor da variável dependente, y .
- A regressão polinomial é um caso particular da regressão linear.



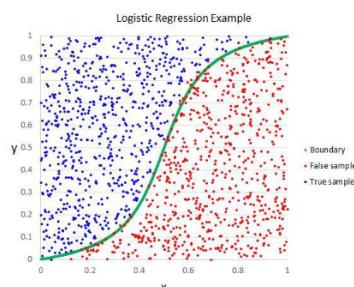
• Regressão Logística

↳ Variável dependente é de natureza categórica

- Em contraste, a **regressão** (linear, múltipla, ...) é usada quando a variável dependente é **contínua** e a natureza da linha de regressão é linear.

- A **Regressão Logística** é uma técnica de **classificação**:

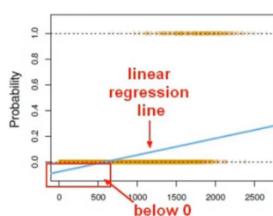
- Empréstimo (SIM/NÃO)
- Diagnóstico (São/Doente)
- Vinho (Branco/Rosé/Tinto)



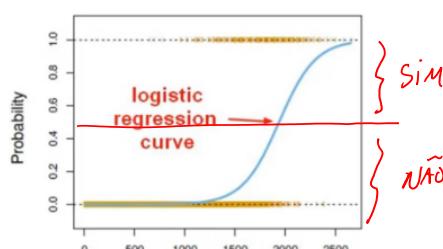
11

- Técnicas de regressão** são usadas, normalmente, para prever uma variável dependente contínua;
- Apesar de a designação poder originar alguma confusão, a **regressão logística** permite resolver problemas de classificação, em que se estimam categorias (valores discretos);

Usar uma função linear de regressão não produz bons resultados na previsão de uma variável binária:



1 Previsão de uma variável binária



AVALIAÇÃO DE MODELOS E MÉTRICAS DE QUALIDADE

Classificação:

- o x : atributo, preditor, variável independente, entrada
- o y : classe, resposta, variável dependente, saída

Tarefa:

- o Aprender um modelo que mapeia cada conjunto de atributos x em um das classes predefinidas de y

Regressão:

Tarefa:

- o Aprender uma equação de reta que analisa variáveis independentes (preço do gás, preço do dólar, custos de transporte), para prever o comportamento de uma variável dependente (preço do petróleo).

Uma Árvore de Decisão pode ser utilizada para fazer regressão:

- o Regressão linear, polinomial, múltipla, entre outras;
- o Prever o preço do petróleo/gás/combustíveis: escala contínua ou real, em € ou \$
- o Estimar a temperatura para o dia de amanhã: escala contínua, em °C ou °F

AVALIAÇÃO

Dados de treino:

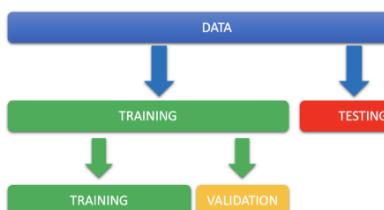
- o Conjunto de dados usado para ajustar o modelo;

Dados de teste:

- o Conjunto de dados usado para fornecer uma avaliação imparcial de um modelo final ajustado ao conjunto de dados de treino.

Dados de validação:

- o Conjunto de dados usado para fornecer uma avaliação imparcial de um ajuste do modelo, no conjunto de dados de treino;



The literature on machine learning often reverses the meaning of "validation" and "test" sets. This is the most blatant example of the terminological confusion that pervades artificial intelligence research.

The crucial point is that a test set, by the standard definition in the NN [neural net] literature, is never used to choose among two or more networks, so that the error on the test set provides an unbiased estimate of the generalization error (assuming that the test set is representative of the population, etc.).

- **Training Dataset:** The sample of data used to fit the model.
- **Validation Dataset:** The sample of data used to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyperparameters. The evaluation becomes more biased as skill on the validation dataset is incorporated into the model configuration.
- **Test Dataset:** The sample of data used to provide an unbiased evaluation of a final model fit on the training dataset.

```
train, validation, test = split(data)
```

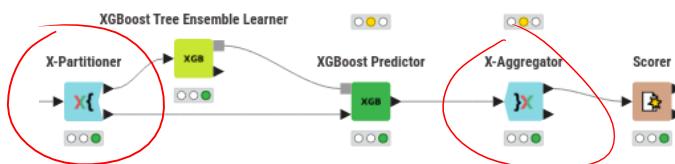
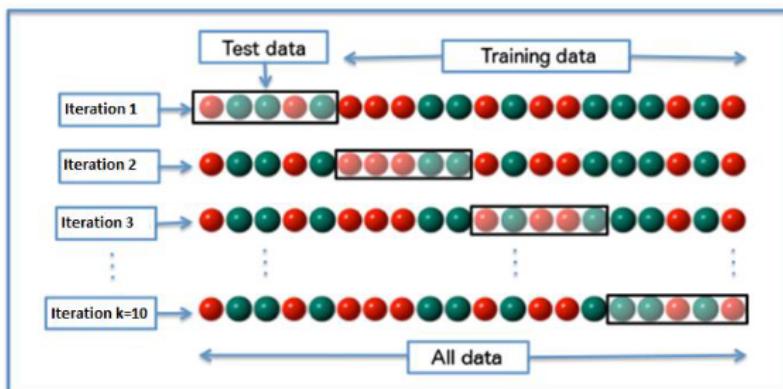
...
...

```
skill = evaluate(model, validation)
```

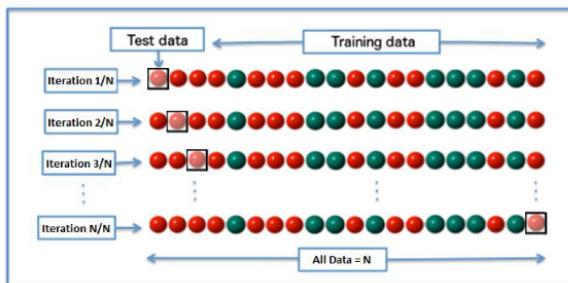
...
...

```
skill = evaluate(model, test)
```

CROSS VALIDATION



• Leave-one-out cross-validation



- Se o *dataset* for grande, um valor pequeno para *k* pode ser suficiente, uma vez que teremos uma quantidade grande de dados para treino;
- Se o *dataset* for pequeno, um valor grande de *k* ≈ *N* pode revelar-se mais adequado para maximizar a quantidade de dados para treino;

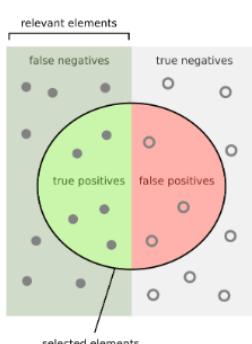
Métricas de qualidade

- Matrizes de Confusão
 - Tabela utilizada para descrever o desempenho de um modelo de classificação.

Accuracy

Quantidade de previsões corretas dividido pela quantidade total de observações:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$



		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

de observações:



$$\text{Precision} = \frac{\text{How many selected items are relevant?}}{\text{How many relevant items are selected?}}$$

$$\text{Recall} = \frac{\text{How many relevant items are selected?}}{\text{How many selected items are relevant?}}$$

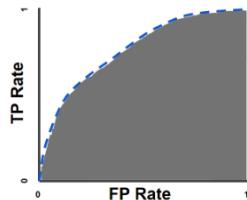
How many selected items are relevant?
How many relevant items are selected?

- Matrizes de Confusão

- Tabela utilizada para descrever o desempenho de um modelo de classificação.

- AUC curve:

- A Area Under the Curve (AUC) mede a área abaixo da curva ROC;
- Mede quanto bem as previsões são classificadas, em vez de avaliar os seus valores absolutos (varia de 0 a 1);
- Um modelo cujas previsões estão 100% erradas tem uma AUC de 0; aquele cujas previsões estão 100% corretas tem uma AUC de 1.



Métricas de qualidade : Regressão

$$MAE = \frac{1}{n} \sum \left| \underbrace{y - \hat{y}}_{\text{The absolute value of the residual}} \right|$$

Divide by the total number of data points
Sum of
Actual output value
Predicted output value

Erro médio absoluto

$$MSE = \frac{1}{n} \sum \left(\underbrace{y - \hat{y}}_{\text{The absolute value of the residual}} \right)^2$$

Divide by the total number of data points
Sum of
Actual output value
Predicted output value

$$RMSE = \sqrt{\sum_{i=1}^n \frac{(\hat{y}_i - y_i)^2}{n}}$$

Raiz quadrada do erro médio quadrado

- Todos variam de 0 a ∞ e são indiferentes à direção dos erros;
- MAE e RMSE expressam o erro de previsão na mesma unidade da variável de interesse;
- MSE e RMSE, ao elevar o erro ao quadrado, dão um peso relativamente alto para erros grandes;
- MSE e RMSE são mais úteis quando grandes erros são especialmente indesejáveis.

APRENDIZAGEM SEM SUPERVISÃO : SEGMENTAÇÃO

- A Segmentação/*Clustering* de dados é um processo através no qual se **particiona** um conjunto de **dados em segmentos/clusters** de menor dimensão, que agrupam conjuntos de dados **similares**.



- Um Segmento/*Cluster* é uma coleção de valores/objetos que:

- são similares entre si, dentro de um mesmo segmento;
- são diferentes dos valores/objetos de outros segmentos:

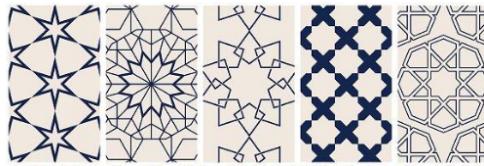


- Medidas de similaridade:

- distância Euclidiana ou de Manhattan, para atributos contínuos;
- coeficiente de Jaccard, para atributos discretos/binários;
- etc.

- A deteção de segmentos é útil:

- quando se suspeita da **existência de agrupamentos "naturais"**, que podem representar grupos de clientes, de produtos ou de bens que partilhem (muita) informação;
- quando existem **muitos padrões diferentes** nos dados, dificultando a tarefa de identificar um determinado padrão;
- a criação de segmentos semelhantes reduz a complexidade do problema.



(MAS) A noção de segmento
é ambígua

- Atributos contínuos:

- normalizar os dados: evita que os resultados dependam das unidades de medida;
- normalmente, utilizam-se medidas de distância para calcular a proximidade (similaridade) entre objetos:

- distância Euclidiana: é a medida de distância geométrica no espaço (a mais usada):

$$d(x, y) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (\text{para 2 dimensões})$$

- distância *Manhattan*: mede a distância pela diferença entre os pontos (função não quadrática):

$$d(x, y) = |x_1 - x_2| + |y_1 - y_2| \quad (\text{para 2 dimensões})$$

- distância *Minkowski*: mede o peso progressivo em função da distância dos pontos:

$$d(i, j) = \left(|x_{i1} - x_{j1}|^p + |x_{i2} - x_{j2}|^p \right)^{\frac{1}{p}} \quad (\text{para } n \text{ dimensões}, c/p \geq 1)$$

(é uma generalização das duas anteriores).

$$d(i, j) = \left(\sum_{k=1}^n |x_{ik} - x_{jk}|^p \right)^{\frac{1}{p}} \quad (\text{para } n \text{ dimensões}, c/p \geq 1)$$

- Atributos binários:

- são classificados em:

- Simétricos**: significado de ser 0 é o mesmo de ser 1;

- Assimétricos**: significado de ser 0 é diferente de ser 1;

- a similaridade calculada com base em atributos simétricos é designada **similaridade invariante**;

no caso oposto diz-se **similaridade não-invariante**;

- tabela de contingência para os dados binários:

- coeficiente simples (simétricos):

$$d(i, j) = \frac{b + c}{a + b + c + d}$$

- coeficiente Jaccard (assimétricos):

$$d(i, j) = \frac{b + c}{a + b + c}$$

	Sexo	Febre	Tosse	Dor
João	M	Sim	Não	Não
Maria	F	Sim	Não	Sim
José	M	Sim	Sim	Não

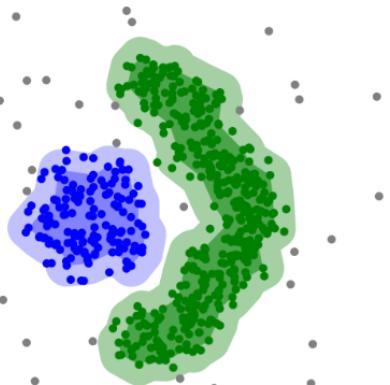
Simétrico Assimétrico

		Maria			
		Sexo	M	F	Soma
João	M	a = 0	b = 1	a+b	
	F	c = 0	d = 0	c+d	
	Soma	a+c	b+d		

		Maria			
		F/T/D	S	N	Soma
João	S	a = 1	b = 0	a+b	
	N	c = 1	d = 1	c+d	
	Soma	a+c	b+d		

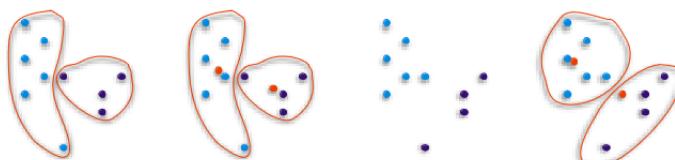
Principais Métodos de Segmentação

- Particionamento:
 - criar várias partições e adotar um critério de avaliação;
- Hierarquização:
 - decompor hierarquicamente o conjunto de dados;
- Outros:
 - Baseados na Densidade:
 - aumentar o segmento enquanto a densidade de pontos estiver num determinado limite (utilizam-se funções de conectividade e densidade);



**Algoritmos de Particionamento
Método k-means**

- Sendo dado 'k' (número de segmentos), seguir os 4 passos:
 1. Dividir os objetos em 'k' subconjuntos não vazios;
 2. Calcular o centro de cada segmento (centroid);
 3. Atribuir cada objeto ao centroid mais próximo;
 4. Voltar ao ponto 2.; parar quando não houver mais possibilidades de atribuição.



**Algoritmos de Particionamento
Método k-means**

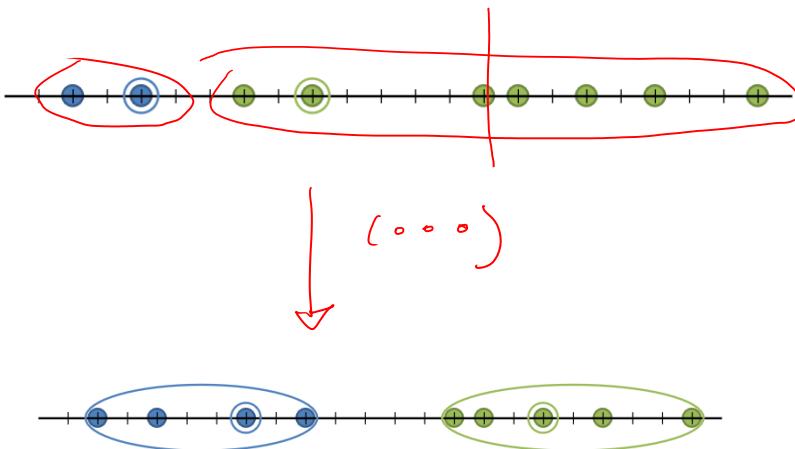
- Vantagens:
 - Relativamente eficiente: sendo 'n' o número de objetos, 'k' o número de segmentos e 'i' o número de iterações, normalmente acontece $k \cdot i \ll n$;
 - Termina com ótimos locais.
- Desvantagens:
 - Aplicável, apenas, quando é possível calcular a média (*mean*);
 - É necessário identificar o número de segmentos *a priori*;
 - Incapacidade de lidar com ruído nos dados;
 - Inadequado para determinar segmentos côncavos.



Algoritmos de Particionamento

Método k-medoids

- Medoids são objetos **representativos** do conjunto de dados;
- Inicia-se com um conjunto de medoids que, iterativamente, vão sendo substituídos por outros não-medoids desde que a distância do segmento resultante seja melhorada.



- Vantagens e Desvantagens:

- É mais robusto do que o método k-means na presença de dados ruidosos, uma vez que os objetos selecionados são menos influenciáveis por valores extremos do que a média (*mean*);
- Produz bons resultados para conjuntos de dados de pequenas dimensões;
- Não se comporta tão bem quando se pretende a sua aplicação em conjuntos de dados de grandes dimensões.

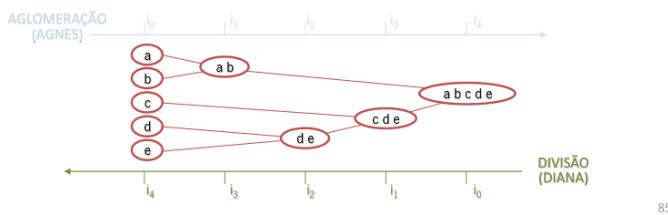
ALGORITMOS DE HIERARQUIZAÇÃO

- Aglomeração:

- Inicia-se formando segmentos com um objeto, para todos os objetos;
- Prossegue juntando segmentos atómicos em segmentos cada vez mais amplos.

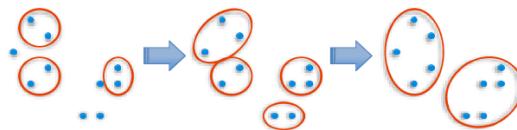
- Divisão:

- Inicia-se com todos os objetos em um só segmento que vai subdividindo em segmentos de menor dimensão;
- Aplicação prática muito rara.



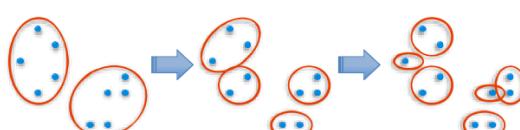
Algoritmos de Hierarquização AGNES: Agglomerative Nesting

- Iterativamente, vai juntando objetos que apresentam menores valores de dissemelhança: os conjuntos C1 e C2 são juntos se os objetos de C1 e de C2 produzem o menor valor de distância Euclidiana entre quaisquer dois objetos de segmentos distintos.



Algoritmos de Hierarquização DIANA: Divisive Analysis

- Iterativamente e partindo de um segmento composto por todos os objetos, dividir em segmentos menores que maximizam a distância Euclidiana entre objetos vizinhos de segmentos diferentes.



Semana 8 : Redes neurais artificiais

Machine Learning

Definição

- "Every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it.

An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves."

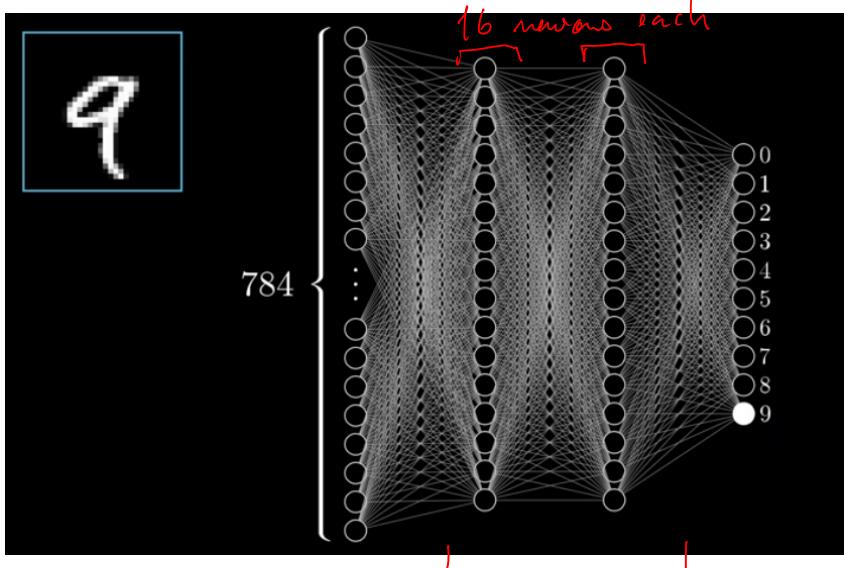
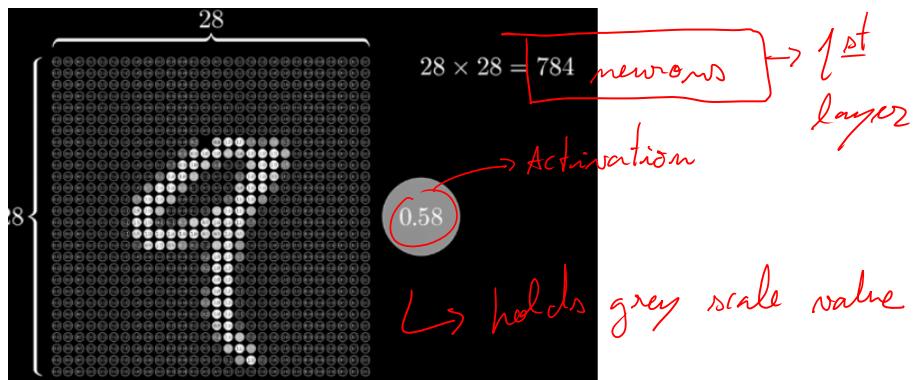
John McCarthy, Dartmouth Conference, 1956
(<http://jmc.stanford.edu>)



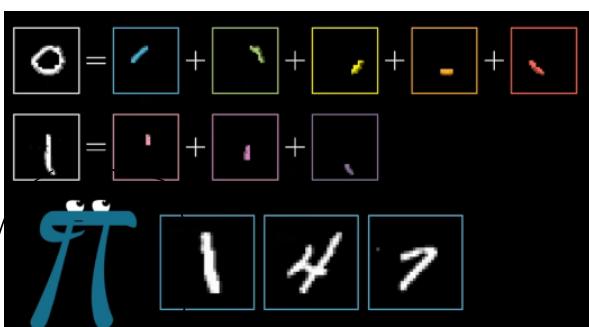
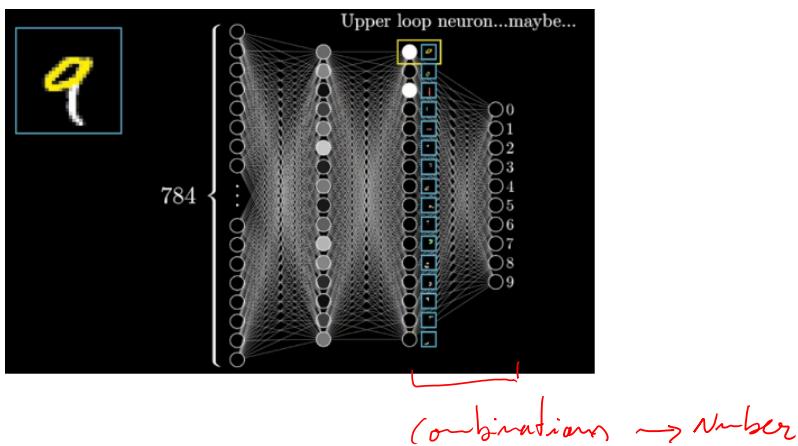
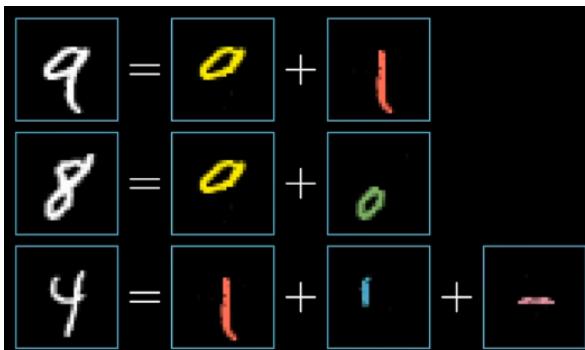
- Cada aspecto da aprendizagem ou qualquer outra característica da inteligência pode, em princípio, ser descrito de forma tão precisa que será possível construir uma máquina para o simular.
Serão feitas tentativas para descobrir como fazer com que as máquinas usem a linguagem, formem abstrações e conceitos, resolvam tipos de problemas até agora reservados para os humanos, e sejam capazes de se melhorarem a si próprias.

0.2

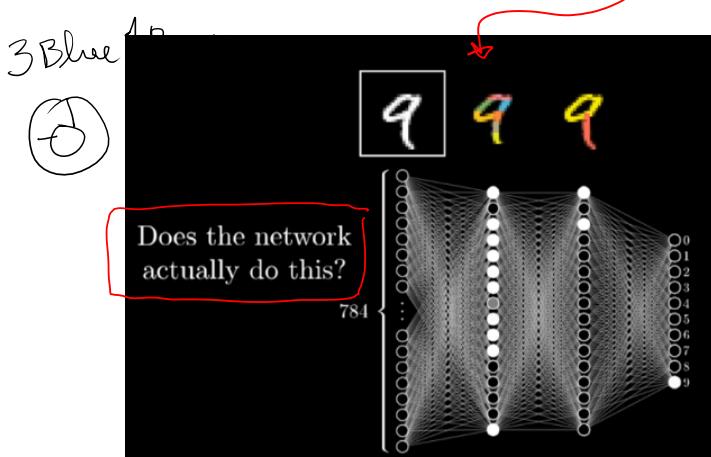
Neuron \rightarrow Thing that holds a number [0, 1]



PATTERNS



$\pi \rightarrow$ Little edge \rightarrow 1^o layer?

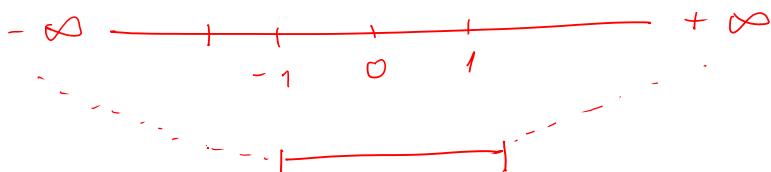
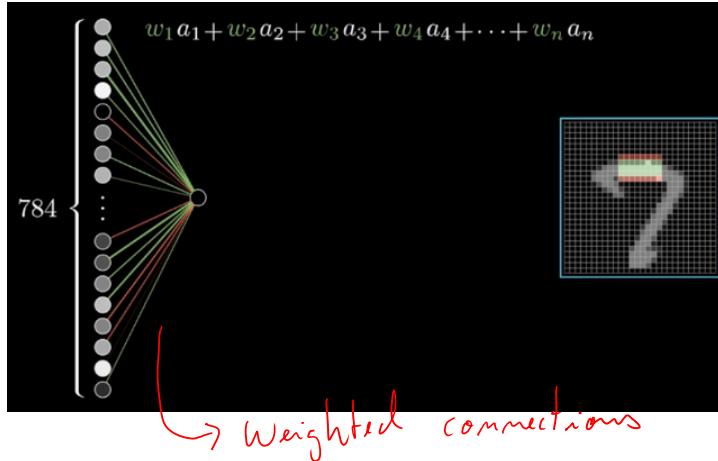


Layers of abstraction

Example:



What are the connections
are actually doing?

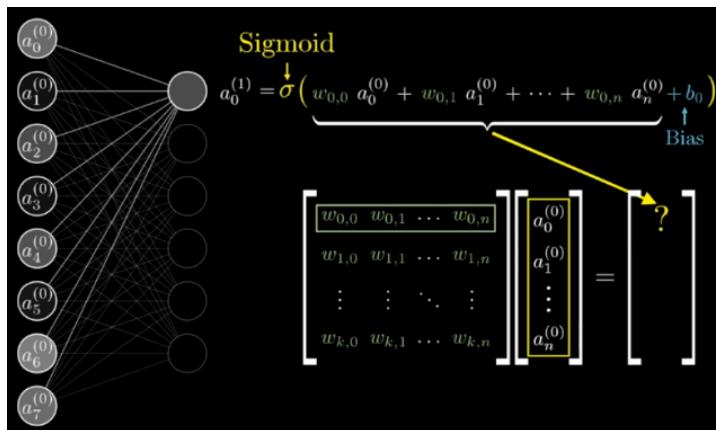


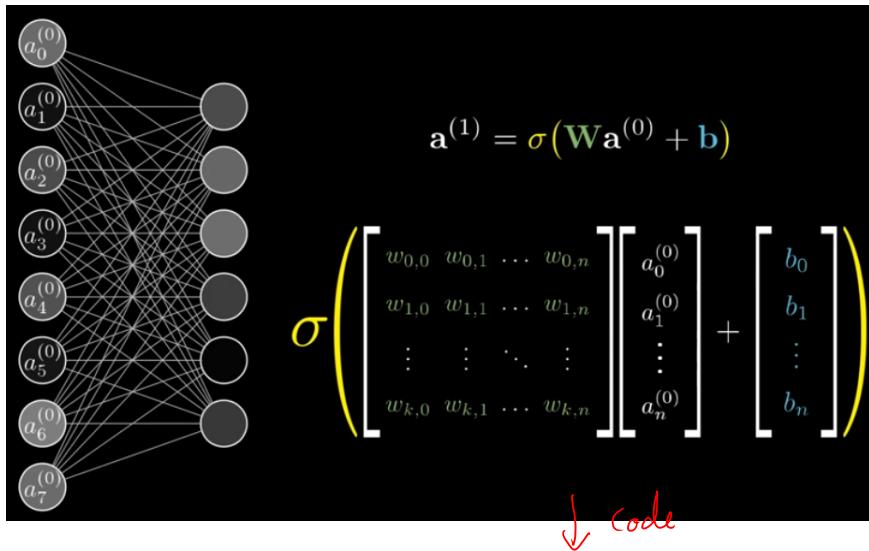
Activations should
be in this range

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

Sigmoid
 \downarrow How positive is this?
 $\sigma(w_1 a_1 + w_2 a_2 + w_3 a_3 + \dots + w_n a_n \boxed{-10})$
 "bias"

⇒ Only active when > 10 ↗ Bias for inactivity





$$\mathbf{a}^{(1)} = \sigma(\mathbf{W}\mathbf{a}^{(0)} + \mathbf{b})$$

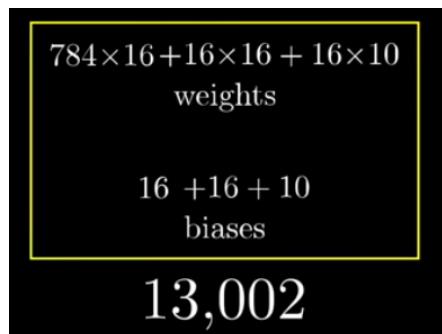
```
class Network(object):
    def __init__(self, *args, **kwargs):
        #...yada yada, initialize weights and biases...

    def feedforward(self, a):
        """Return the output of the network for an input vector a"""
        for b, w in zip(self.biases, self.weights):
            a = sigmoid(np.dot(w, a) + b)
        return a
```

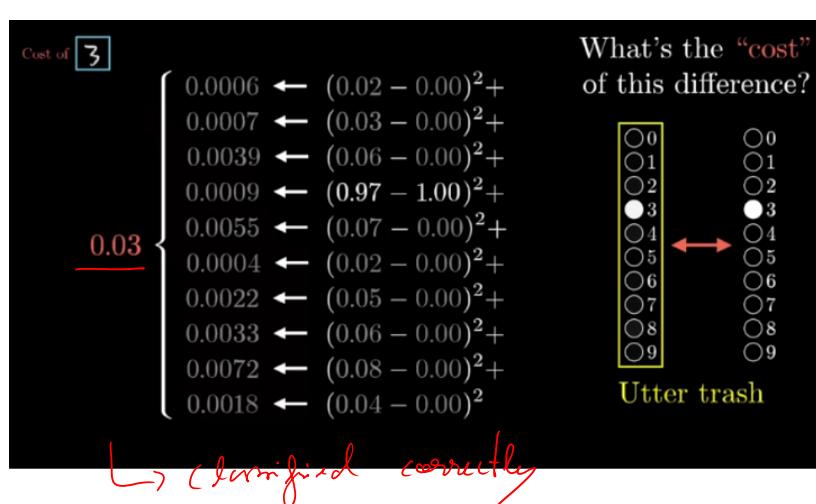
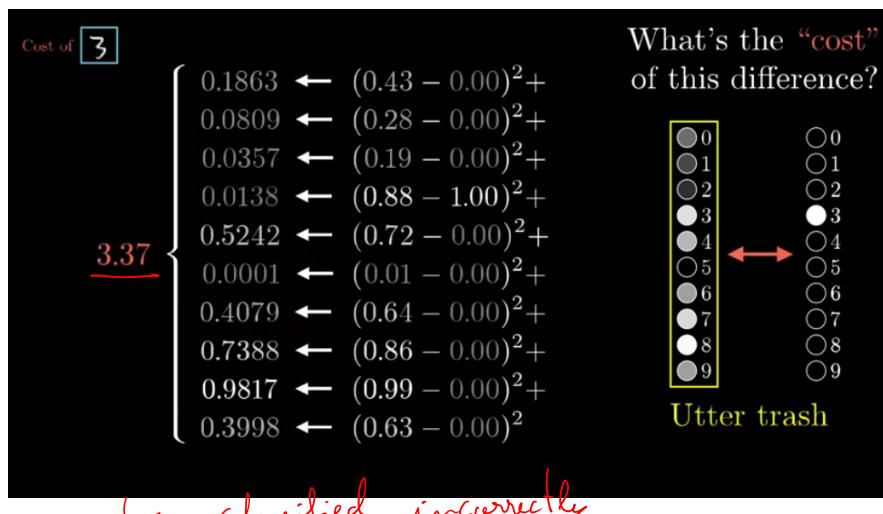
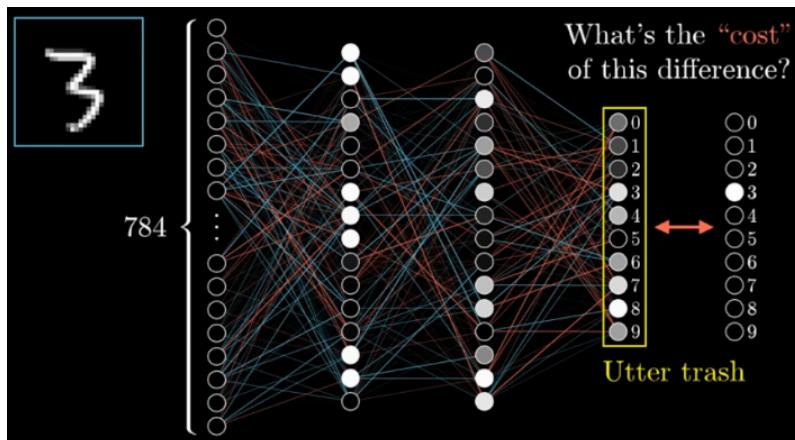
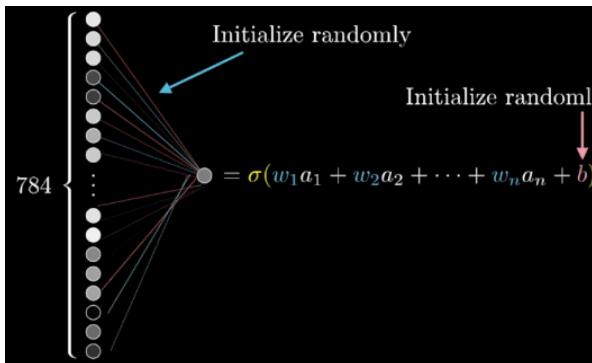
A NEURON
IS A FUNCTION



WEIGHTS



LEARNING : Finding the right
weights and values



Neural network function

Input: 784 numbers (pixels)

Output: 10 numbers

Parameters: 13,002 weights/biases

Cost function

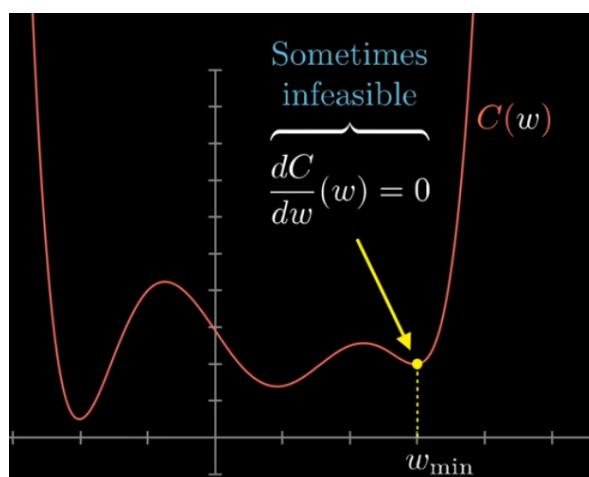
Input: 13,002 weights/biases

Output: 1 number (the cost)

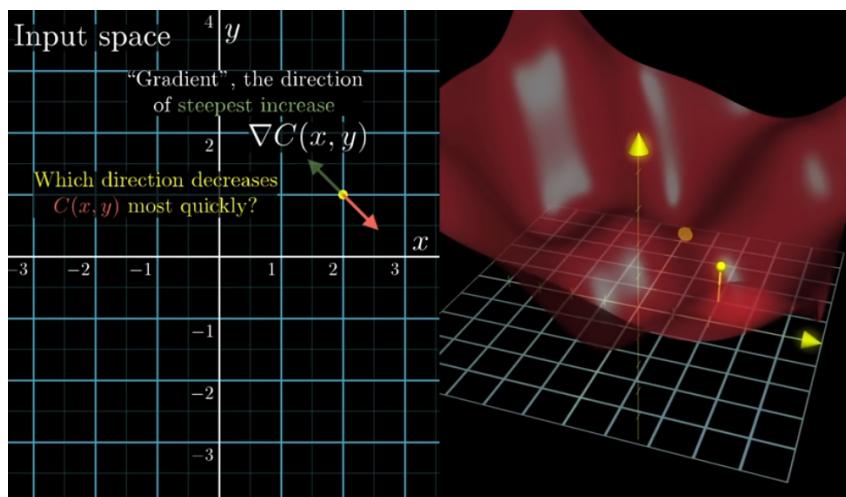
Parameters: Many, many, many training examples

$$\left(\boxed{3}, 3 \right)$$

How to tell the network how to get better?



↳ MNM
Find local mins



Learning: minimizing a cost function

13,002 weights and biases

How to nudge all weights and biases

$$\vec{\mathbf{W}} = \begin{bmatrix} 2.43 \\ -1.57 \\ 1.98 \\ \vdots \\ -1.16 \\ 3.82 \\ 1.21 \end{bmatrix} \quad -\nabla C(\vec{\mathbf{W}}) = \begin{bmatrix} 0.18 \\ 0.45 \\ -0.51 \\ \vdots \\ 0.40 \\ -0.32 \\ 0.82 \end{bmatrix}$$

negative gradient
of the cost function

Which nudges is
gonna cause the
most rapid decrease
to the cost function

runs Backpropagation

$$\vec{\mathbf{W}} = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_{13,000} \\ w_{13,001} \\ w_{13,002} \end{bmatrix}$$

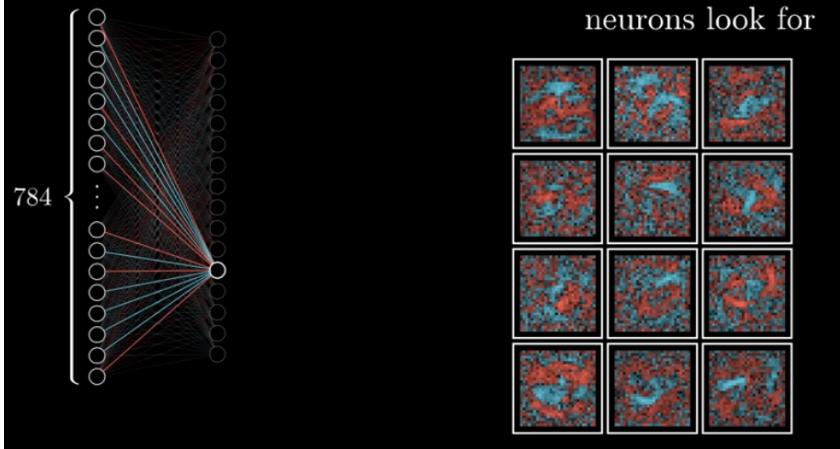
$$-\nabla C(\vec{\mathbf{W}}) = \begin{bmatrix} 0.31 \\ 0.03 \\ -1.25 \\ \vdots \\ 0.78 \\ -0.37 \\ 0.16 \end{bmatrix}$$

w₀ should increase somewhat
w₁ should increase a little
w₂ should decrease a lot
w_{13,000} should increase a lot
w_{13,001} should decrease somewhat
w_{13,002} should increase a little

gradient of the
cost function

→ which weights
matter the most

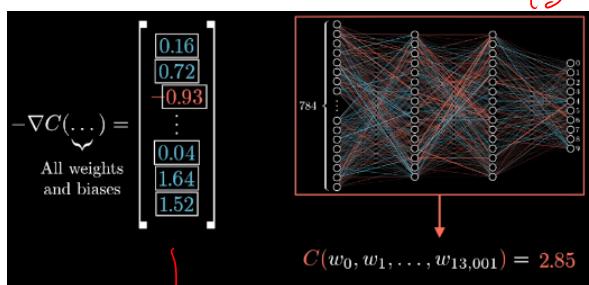
What second layer
neurons look for



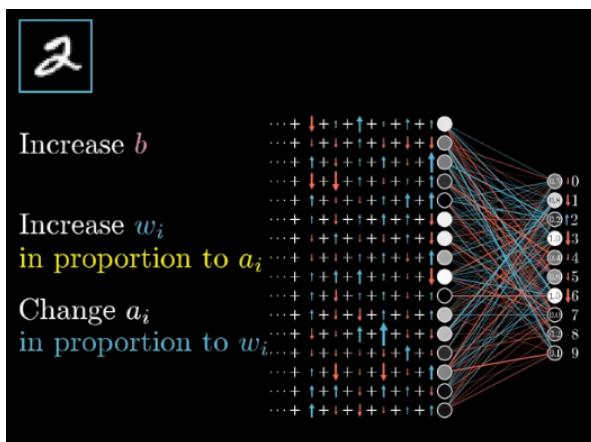
BACK PROPAGATION

↳ algorithm for computing
the gradient

↳ direction in
13 002 dimensions



↳ How sensitive the cost
function is to which weight
and bias

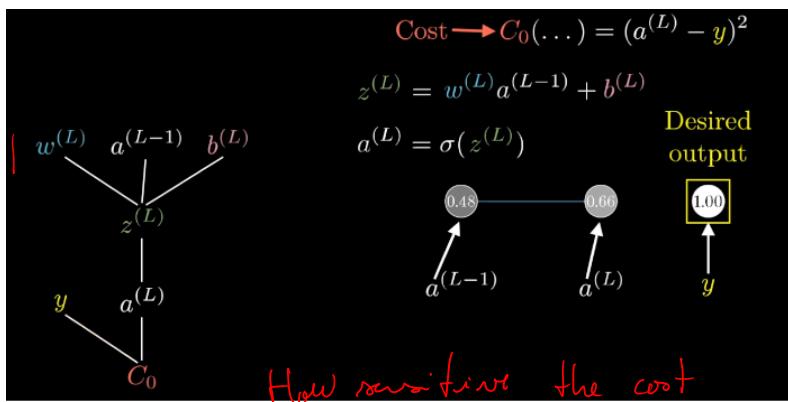


	2	5	0	4	1	9	...	Average over all training data
w_0	-0.08	+0.02	-0.02	+0.11	-0.05	-0.14	...	-0.08
w_1	-0.11	+0.11	+0.07	+0.02	+0.09	+0.05	...	+0.12
w_2	-0.07	-0.04	-0.01	+0.02	+0.13	-0.15	...	-0.06
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\ddots	\vdots
$w_{13,001}$	+0.13	+0.08	-0.06	-0.09	-0.02	+0.04	...	+0.04

negative gradient
of the cost function

Computationally slow
 \Rightarrow create mini-batches

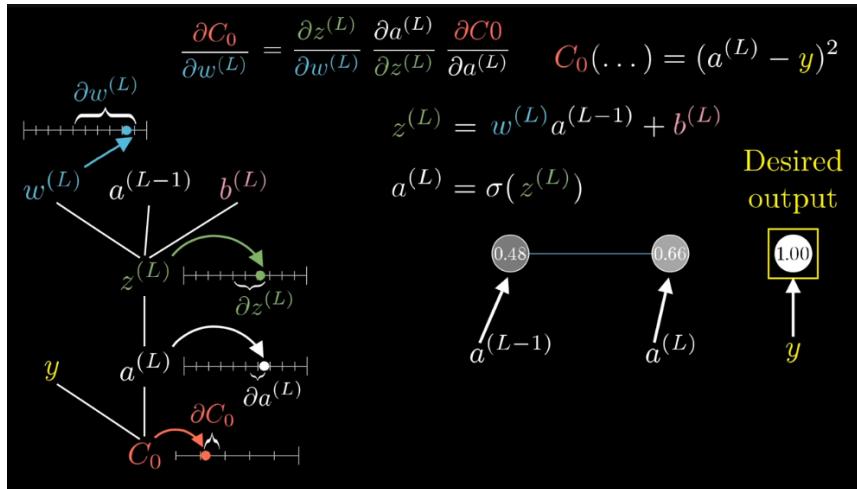
Backpropagation Calculus



↓

$$\frac{\partial C_0}{\partial w^{(L)}}$$

Chain rule : (For the weight)



$$\frac{\partial C_0}{\partial a^{(L)}} = 2(a^{(L)} - y)$$

$$C_0 = (a^{(L)} - y)^2$$

$$\frac{\partial a^{(L)}}{\partial z^{(L)}} = \sigma'(z^{(L)})$$

$$a^{(L)} = \sigma(z^{(L)})$$

$$\boxed{\frac{\partial z^{(L)}}{\partial w^{(L)}}} = a^{(L-1)}$$

$$z^{(L)} = w^{(L)}a^{(L-1)} + b^{(L)}$$

A variação do peso
depende fortemente do
quão forte o neurônio
anterior é

"Neurons that fire together
wire together"

Average of all training examples

$$\underbrace{\frac{\partial C}{\partial w^{(L)}}}_{\text{Derivative of full cost function}} = \overbrace{\frac{1}{n} \sum_{k=0}^{n-1} \frac{\partial C_k}{\partial w^{(L)}}}^{\text{Derivative of full cost function}}$$

$$\frac{\partial C_0}{\partial w_{jk}^{(L)}} = \frac{\partial z_j^{(L)}}{\partial w_{jk}^{(L)}} \frac{\partial a_j^{(L)}}{\partial z_j^{(L)}} \frac{\partial C_0}{\partial a_j^{(L)}}$$

$$z_j^{(L)} = \dots + w_{jk}^{(L)} a_k^{(L-1)} + \dots$$

$$a_j^{(L)} = \sigma(z_j^{(L)})$$

$$\frac{\partial C_0}{\partial a_k^{(L-1)}} = \underbrace{\sum_{j=0}^{n_L-1} \frac{\partial z_j^{(L)}}{\partial a_k^{(L-1)}} \frac{\partial a_j^{(L)}}{\partial z_j^{(L)}} \frac{\partial C_0}{\partial a_j^{(L)}}}_{\text{Sum over layer L}}$$

$$z_j^{(L)} = \dots + w_{jk}^{(L)} a_k^{(L-1)} + \dots$$

$$a_j^{(L)} = \sigma(z_j^{(L)})$$

$$C_0 = \sum_{j=0}^{n_L-1} (a_j^{(L)} - y_j)^2$$

