



# IA353A - Neural Networks EC1

Rafael Claro Ito  
(R.A.: 118430)

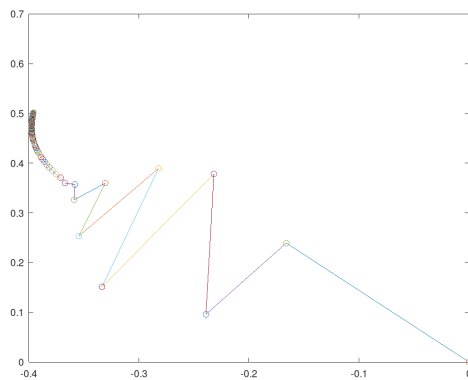
May 2020

# Question 1

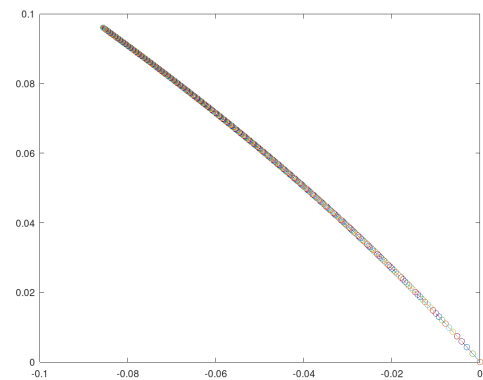
## 1.1 SGD

### 1.1.1 Underdetermined

```
1 % SGD para o caso de sistema linear subdeterminado
2 clear all;
3 randn('state',0);
4 N = 10;
5 Nit = 500;
6 X = randn(N,2*N);
7 S = sign(randn(N,1));
8 w = (X'/(X*X'))*S;
9 w1 = zeros(2*N,1);
10 %passo = 0.1;
11 passo = 0.001;
12 for it=2:Nit,
13     w1(:,it) = w1(:,it-1) - (passo/sqrt(it))*(X'*X*w1(:,it-1)-X'*S);
14 end
15 figure(1);
16 title('Stochastic Gradient Descent');
17 for it = 1:(Nit-1),
18     plot([w1(1,it);w1(1,it+1)],[w1(2,it);w1(2,it+1)]);hold on;
19     plot(w1(1,it),w1(2,it),'o');
20     plot(w1(1,it+1),w1(2,it+1),'o');
21 end
22 hold off;
23 disp('[Minimum Norm Solution Obtained solution]');
24 disp([w w1(:,Nit)]);
25 [S X*w X*w1(:,Nit)]
26 %-----
27 % save figure
28 path = strcat(' ../figures/Q1/sgd_under_step_', num2str(passo), '.png');
29 saveas(gcf, path);
```



(a) Progression of W using step of 0.1



(b) Progression of W using step of 0.001

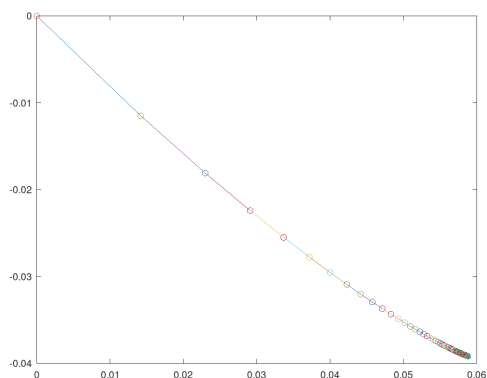
Figure 2: Progression of W for underdetermined system using SGD

### 1.1.2 Overdetermined

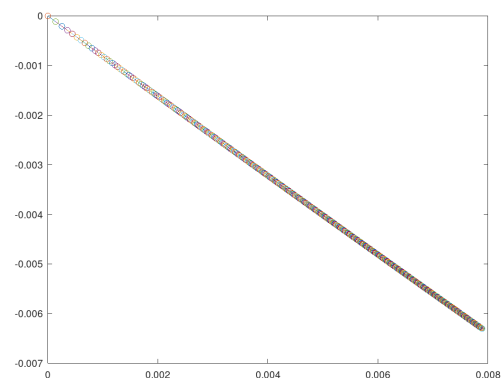
```

1 % SGD para o caso de sistema linear sobredeterminado
2 clear all;
3 randn('state',0);
4 N = 10;
5 Nit = 500;
6 X = randn(N,2);
7 S = sign(randn(N,1));
8 w = (X'*X)\X'*S;
9 disp('Optimal solution');
10 disp(w);
11 w1 = zeros(2,1);
12 %passo = 0.1;
13 passo = 0.001;
14 for it=2:Nit,
15     w1(:,it) = w1(:,it-1) - (passo/sqrt(it))*(X'*X*w1(:,it-1)-X'*S);
16 end
17 figure(1);
18 title('Stochastic Gradient Descent');
19 for it = 1:(Nit-1),
20     plot([w1(1,it);w1(1,it+1)],[w1(2,it);w1(2,it+1)]);hold on;
21     plot(w1(1,it),w1(2,it),'o');
22     plot(w1(1,it+1),w1(2,it+1),'o');
23 end
24 hold off;
25 disp('Obtained solution');
26 disp(w1(:,Nit));
27 [S X*w X*w1(:,Nit)]
28 %-----
29 % save figure
30 path = strcat(' ../figures/Q1/sgd_over_step_', num2str(passo), '.png');
31 saveas(gcf, path);

```



(a) Progression of W using step of 0.1



(b) Progression of W using step of 0.001

Figure 3: Progression of W for overdetermined system using SGD

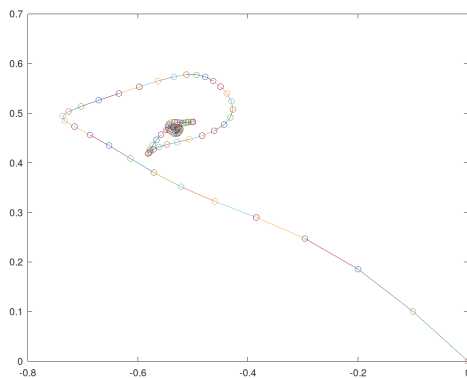
## 1.2 Adam

### 1.2.1 Underdetermined

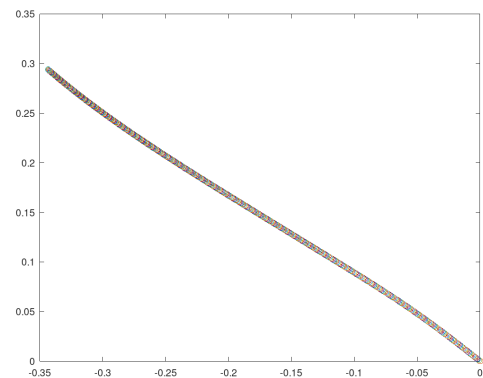
```

1 % Adam para o caso de sistema linear subdeterminado
2 clear all;
3 randn('state',0);
4 N = 10;
5 Nit = 500;
6 X = randn(N,2*N);
7 S = sign(randn(N,1));
8 w = (X'/(X*X'))*S;      % optimal solution
9 w1 = zeros(2*N,1);      % initial weights
10 passo = 0.1;
11 passo = 0.001
12 %-----
13 % parameters
14 beta_1 = 0.9;
15 beta_2 = 0.999;
16 e = 1e-8;
17 m = v = m_hat = v_hat = zeros(2*N,1);
18 %-----
19 % loop
20 for it=2:Nit,
21     g = X'*X*w1(:,it-1)-X'*S;
22     m = beta_1*m + (1-beta_1)*g;
23     v = beta_2*v + (1-beta_2)*g.^2;
24     m_hat = m / (1 - beta_1^(it-1));
25     v_hat = v / (1 - beta_2^(it-1));
26     w1(:,it) = w1(:,it-1) - passo./(sqrt(v_hat) + e) .* m_hat;
27     %w1(:,it) = w1(:,it-1) - (passo/sqrt(it))*(X'*X*w1(:,it-1)-X'*S);
28 end
29 %-----
30 % plot weights
31 figure(1);
32 title('Stochastic Gradient Descent');
33 for it = 1:(Nit-1),
34     plot([w1(1,it);w1(1,it+1)],[w1(2,it);w1(2,it+1)]);hold on;
35     plot(w1(1,it),w1(2,it),'o');
36     plot(w1(1,it+1),w1(2,it+1),'o');
37 end
38 hold off;
39 disp(' [Minimum Norm Solution Obtained solution] ');
40 disp([w w1(:,Nit)]);
41 [S X*w X*w1(:,Nit)]
42 %-----
43 % save figure
44 path = strcat(' ../figures/Q1/adam_under_step_', num2str(passo), '.png');
45 saveas(gcf, path);

```



(a) Progression of W using step of 0.1



(b) Progression of W using step of 0.001

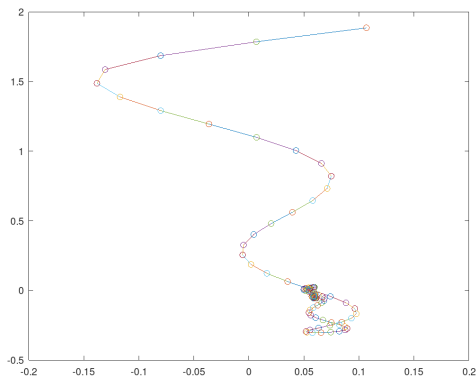
Figure 4: Progression of W for underdetermined system using Adam

### 1.2.2 Overdetermined

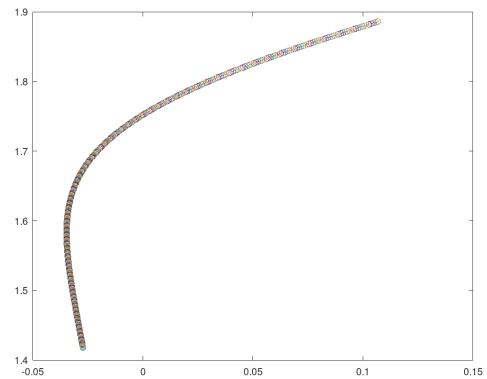
```

1 % Adam para o caso de sistema linear sobredeterminado
2 clear all;
3 randn('state',0);
4 N = 10;
5 Nit = 500;
6 X = randn(N,2);
7 S = sign(randn(N,1));
8 w = (X'*X)\X'*S; % optimal solution
9 disp('Optimal solution');
10 disp(w);
11 w1 = randn(2,1);
12 passo = 0.1;
13 %passo = 0.001
14 %-----
15 % parameters
16 beta_1 = 0.9;
17 beta_2 = 0.999;
18 e = 1e-8;
19 m = v = m_hat = v_hat = zeros(2,1);
20 %-----
21 % loop
22 for it=2:Nit,
23     g = X'*X*w1(:,it-1)-X'*S;
24     m = beta_1*m + (1-beta_1).*g;
25     v = beta_2*v + (1-beta_2).*g.^2;
26     m_hat = m / (1 - beta_1^(it-1));
27     v_hat = v / (1 - beta_2^(it-1));
28     w1(:,it) = w1(:,it-1) - passo./(sqrt(v_hat) + e) .* m_hat;
29     %w1(:,it) = w1(:,it-1) - (passo/sqrt(it))*(X'*X*w1(:,it-1)-X'*S);
30 end
31 %-----
32 % plot weights
33 figure(1);
34 title('Stochastic Gradient Descent');
35 for it = 1:(Nit-1),
36     plot([w1(1,it);w1(1,it+1)],[w1(2,it);w1(2,it+1)]);hold on;
37     plot(w1(1,it),w1(2,it),'o');
38     plot(w1(1,it+1),w1(2,it+1),'o');
39 end
40 hold off;
41 disp('Obtained solution');
42 disp(w1(:,Nit));
43 [S X*w X*w1(:,Nit)]
44 %-----
45 % save figure
46 path = strcat(' ../figures/Q1/adam_over_step_', num2str(passo), '.png');
47 saveas(gcf, path);

```



(a) Progression of W using step of 0.1



(b) Progression of W using step of 0.001

Figure 5: Progression of W for overdetermined system using Adam

### 1.3 Comparison

We can see from the plots that for SGD both underdetermined and overdetermined systems converged to the right solution using step equal to 0.1. When using step equal to 0.001, the optimizer did not converged, but we can see that it is on the way, not reaching it possibly because of the step value.

The same thing happened when using Adam optimizer. For steps equal to 0.1, both underdetermined and overdetermined converged to the right solution, what didn't happen when using step of 0.001.

Comparing both plots, in general we can see that SGD has a smoother path, walking directly to the solution. Adam, on the other hand, due to the adaptative moment estimation, shows a curlier path through the solution, sometimes even surrounding it.

## Question 2

$$\min \|Ax - b\|_P^2 + \|x - x_0\|_Q^2$$

Since we are searching for  $x$  that minimizes the previous expression, we will calculate the derivative with relation to  $x$  and set it equal to zero:

$$\frac{d}{dx}(\|Ax - b\|_P^2 + \|x - x_0\|_Q^2) = 0$$

**Property used:**  $\|x\|_Q^2 = x^T Q x$

$$\frac{d}{dx} \left[ \overbrace{(Ax - b)^T P (Ax - b)}^{\|Ax - b\|_P^2} + \overbrace{(x - x_0)^T Q (x - x_0)}^{\|x - x_0\|_Q^2} \right] = 0$$

**Property used:**  $(M + N)^T = M^T + N^T$

$$\frac{d}{dx} \{ [(Ax)^T - b^T] P (Ax - b) + (x^T - x_0^T) Q (x - x_0) \} = 0$$

**Property used:**  $(MN)^T = N^T M^T$

$$\frac{d}{dx} \{ [x^T A^T - b^T] P (Ax - b) + (x^T - x_0^T) Q (x - x_0) \} = 0$$

$$\frac{d}{dx} [(x^T A^T P A x - x^T A^T P b - b^T P A x + b^T P b) + (x^T Q x - x^T Q x_0 - x_0^T Q x + x_0^T Q x_0)] = 0$$

**Properties used:**

- $\frac{d}{dy}(y^T M y) = M^T y + M y$ .
- $\frac{d}{dy}(y^T M y) = 2M y$ , if  $M = M^T$  (i.e.  $M$  is symmetric)
- $\frac{d(Ax)}{dx} = A$
- $\frac{d(x^T A)}{dx} = A^T$
- obs.:  $A^T P A$  is symmetric, since  $(A^T P A)^T = (P A)^T (A^T)^T = A^T P^T A$ , but  $P^T = P$ , since  $P$  is symmetric. Then  $A^T P A = (A^T P A)^T$

**Using the previous properties, we have:**

$$\frac{d}{dx} [(x^T A^T P A x - x^T A^T P b - b^T P A x + b^T P b) + (x^T Q x - x^T Q x_0 - x_0^T Q x + x_0^T Q x_0)] = 0$$

$$[2A^T P A x - (A^T P b)^T - b^T P A + 0] + [2Q x - (Q x_0)^T - x_0^T Q + 0] = 0$$

$$2A^T P A x - (P b)^T (A^T)^T - b^T P A + 2Q x - x_0^T Q^T - x_0^T Q = 0$$

$$2A^T P A x - b^T P^T A - b^T P A + 2Q x - x_0^T Q - x_0^T Q = 0$$

$$2A^T P A x - b^T P A - b^T P A + 2Q x - x_0^T Q - x_0^T Q = 0$$

$$2A^T P A x - 2b^T P A + 2Q x - 2x_0^T Q = 0$$

$$2A^T P A x + 2Q x = 2b^T P A + 2x_0^T Q$$

$$A^T P A x + Q x = b^T P A + x_0^T Q$$

$$(A^T P A + Q)^{-1} (A^T P A + Q) x = (A^T P A + Q)^{-1} (b^T P A + x_0^T Q)$$

$$x = (A^T P A + Q)^{-1} (b^T P A + x_0^T Q)$$

$$x = (A^T P A + Q)^{-1} (b^T P^T A + x_0^T Q^T)$$

$$x = (A^T P A + Q)^{-1} [(P b)^T A + (Q x_0)^T]$$

$$x = (A^T P A + Q)^{-1} [(A^T P b)^T + (Q x_0)^T]$$

$$\boxed{x = (A^T P A + Q)^{-1} (A^T P b + Q x_0)^T}$$

If the regularization coefficient was used, we would have:

$$\min \|Ax - b\|_P^2 + \alpha^2 \|x - x_0\|_Q^2$$

This, would lead to the solution:

$$x = (A^T P A + \alpha^2 Q)^{-1} (A^T P b + \alpha^2 Q x_0)^T$$

So, even if we choose an  $\alpha$  different than 1, it is always possible to embed its value in the matrix  $Q$ , since  $\alpha$  is only multiplying the  $Q$  matrix.



### Question 3

a)

If  $M$  is symmetric, then  $M^T$  must be equal to  $M$ :

$$M^T = (N^T N)^T = N^T (N^T)^T = N^T N = M$$

Now we need to prove that  $M$  must be positive semi-definite, i.e.,  $z^T M z \geq 0, \forall z \in R^n$ :

$$z^T M z = z^T (N^T N) z = (z^T N^T) (N z) = (N z)^T (N z) = \|N z\|^2 \geq 0$$

b)

Since  $N$  has full rank, we have  $\min(m, n) = n$  linearly independent columns.

$$z^T M z = z^T N^T N z = \|N z\|^2 > 0, \text{ for all } z \neq 0$$

c)

We want to prove that  $N$  may not be unique in  $M = N^T N$ . For this, let's consider the orthogonal matrix  $Q$  and the new matrix that is the result by its multiplication with  $N$ , i.e.,  $QN$ :

$$M = (QN)^T (QN) = (N^T Q^T) (QN) = N^T (Q^T Q) N$$

Since  $Q$  is orthogonal,  $Q^T Q$  is equal to the identity matrix  $I$ . Hence:

$$M = N^T (Q^T Q) N = N^T I N = N^T N$$

So if  $M$  can be decomposed in  $N^T N$ , it can also be decomposed by  $(QN)^T (QN)$ , with  $Q$  being an orthogonal matrix.

d)

In order to find  $N$  we can use Cholesky decomposition. Here we will show an example for  $n = 3$ , but we will generalize to other dimensions:

$$\begin{aligned} M = N^T N &= \begin{pmatrix} N_{11} & 0 & 0 \\ N_{21} & N_{22} & 0 \\ N_{31} & N_{32} & N_{33} \end{pmatrix} \begin{pmatrix} N_{11} & N_{21} & N_{31} \\ 0 & N_{22} & N_{32} \\ 0 & 0 & N_{33} \end{pmatrix} \\ &= \begin{pmatrix} N_{11}^2 & N_{21}N_{11} & N_{31}N_{11} \\ N_{21}N_{11} & N_{21}^2 + N_{22}^2 & N_{31}N_{21} + N_{32}N_{22} \\ N_{31}N_{11} & N_{31}N_{21} + N_{32}N_{22} & N_{31}^2 + N_{32}^2 + N_{33}^2 \end{pmatrix} \quad (\text{symmetric}) \end{aligned}$$

Comparing to the elements of  $M$  we can find  $N$ :

$$N = \begin{pmatrix} \sqrt{M_{11}} & 0 & 0 \\ M_{21}/N_{11} & \sqrt{M_{22} - N_{21}^2} & 0 \\ M_{31}/N_{11} & (M_{32} - N_{31}N_{21})/N_{22} & \sqrt{M_{33} - N_{31}^2 - N_{32}^2} \end{pmatrix}$$

Generalizing for other dimensions, we have:

$$N_{j,j} = (\pm) \sqrt{M_{j,j} - \sum_{k=1}^{j-1} N_{j,k}^2}$$

for elements at the diagonal

$$N_{i,j} = \frac{1}{N_{j,j}} \left( M_{i,j} - \sum_{k=1}^{j-1} N_{i,k} N_{j,k} \right)$$

for  $i > j$

## Question 4

The Taylor series expansion of a function  $f(x, y)$  in a neighborhood around  $(x_0, y_0)$  is as follows:

$$f(x, y) \approx f(x_0, y_0) + \underbrace{f_x(x_0, y_0)(x - x_0) + f_y(x_0, y_0)(y - y_0)}_{\text{first order term}} + \dots$$

Ignoring terms from second order and higher, and considering the linearized function given  $f_L(x, y) = 2x + py - 8$ , we have:

$$f_L(x, y) = f(x_0, y_0) + f_x(x_0, y_0)(x - x_0) + f_y(x_0, y_0)(y - y_0)$$

Taking  $f(x, y) = x\sqrt{y}$ , we have:

- $f_x(x, y) = \sqrt{y}$
- $f_y(x, y) = \frac{x}{2\sqrt{y}}$

$$2x + py - 8 = \underbrace{x_0\sqrt{y_0}}_{f(x_0, y_0)} + \underbrace{\sqrt{y_0}}_{f_x(x_0, y_0)}(x - x_0) + \underbrace{\frac{x_0}{2\sqrt{y_0}}}_{f_y(x_0, y_0)}(y - y_0)$$

$$2x + py - 8 = x_0\sqrt{y_0} + x\sqrt{y_0} - x_0\sqrt{y_0} + \frac{x_0 y}{2\sqrt{y_0}} - \frac{x_0 y_0}{2\sqrt{y_0}}$$

$$2x + py - 8 = \sqrt{y_0}x + \left(\frac{x_0}{2\sqrt{y_0}}\right)y - \left(\frac{x_0 y_0}{2\sqrt{y_0}}\right)$$

Now, if we compare the terms that only depend on  $x$ , that only depend on  $y$  and the independent terms (that only depend on  $x_0$  and  $y_0$ ), we have:

- (i)  $2 = \sqrt{y_0} \implies \boxed{y_0 = 4}$
- (ii)  $p = \left(\frac{x_0}{2\sqrt{y_0}}\right)$
- (iii)  $8 = \left(\frac{x_0 y_0}{2\sqrt{y_0}}\right)$

Substituting in (i) in (iii):

$$\frac{x_0 y_0}{2\sqrt{y_0}} = 8$$

$$\frac{4x_0}{2 \cdot 2} = 8$$

$$\boxed{x_0 = 8}$$

Substituting in (ii):

$$p = \frac{x_0}{2\sqrt{y_0}}$$

$$p = \frac{8}{2 \cdot 2}$$

$$\boxed{p = 2}$$

So the point from where the function was linearized is  $(x_0, y_0) = (8, 4)$  and the coefficient  $p = 2$ , then  $f_L(x, y) = 2x + 2y - 8$ .

## Question 5

Obs.: This question will be answered in Portuguese.

### Dedução:

- Explicação:

Na inferência dedutiva a conclusão sempre está contida nas premissas. Ao derivar uma afirmação  $y$  partindo de  $x$ , dizemos que  $y$  é uma consequência lógica de  $x$ , ou seja, a inferência de  $y$  é baseada no que foi assumido em  $x$  e será sempre verdadeira. Também podemos incluir casos particulares que são inferidos a partir de premissas universais e gerais.

- Vantagem:

Dos três tipos de inferência esta é a única cuja conclusão lógica é sempre verdadeira.

- Desvantagem:

É a única das inferências que não acrescenta conhecimento novo.

- Exemplo:

Se um pesquisador estiver treinando uma rede neural e por algum motivo as informações referentes ao estado atual do treinamento seja perdido (seja por falta de energia, Colab retomando a máquina virtual, ou qualquer outro) sem que haja um backup dos pesos e parâmetros, será necessário reiniciar o treinamento do zero. Logo, o pesquisador opta por salvar checkpoints do status do treinamento (pesos e outros parâmetros), para caso isso ocorra, o treinamento não seja inteiramente perdido e o pesquisador possa continuar do último estado salvo. Essa ação preventiva é tomada baseada em uma inferência dedutiva.

1. Se cai a energia e não tenho backup  $\Rightarrow$  perco o treinamento

2. Quando perco o treinamento (sem backup)  $\Rightarrow$  recomeço do zero

conclusão: se cai energia sem backup, perco tempo recomeçando do zero.

precaução: vou salvar checkpoints do treinamento.

### Indução:

- Explicação:

Diferente da dedução aqui não temos mais a garantia de veracidade a cerca de inferência. Em indução temos a intuição de probabilidade, agregando experiência e conhecimento. A partir de um certo conjunto de elementos conhecidos ou observáveis, e portanto de casos particulares, chega-se em uma conclusão geral. Assim, uma característica marcante deste tipo de inferência é a capacidade de generalização.

- Vantagem:

Entre as vantagens deste modo de inferência podemos listar a capacidade de generalização mesmo sem haver uma certeza lógica e uso de experiências anteriores a cerca do que se está inferindo.

- Desvantagem:

Esta inferência não é 100% verdade como a dedutiva, portanto pode levar a generalizações errôneas. Por trabalhar com experiências passadas, tem potencial de acrescentar informação menor do que a abdução, pois boa parte dela foi usada como experiência para a própria inferência.

- Exemplo:

Ao rodar o loop de treinamento de uma rede neural, o pesquisador se depara com uma exceção Python por conta de um erro em operações envolvendo matrizes que apresenta um log de erro não direto e difícil de interpretar, não sendo muito útil portanto. Em experiências anteriores, na maior parte das vezes o erro era causado por descasamento nos shapes de tensores/arrays ou dimensões erradas. Assim, o pesquisador decide usar um debugger (ex: pdb) colocando o breakpoint imediatamente na linha antes do erro acontecer e checar os shapes e dimensões dos tensores envolvidos, pois desconfia que o erro envolva uma dessas duas coisas.

## Abdução:

- Explicação:

Na inferência abdutiva trabalha-se com hipóteses para explicação do que foi observado. Trata-se de inferir  $x$  como explicação para  $y$ . A conclusão não é dado pela lógica, mas sim pela capacidade em justificar e argumentar a escolha da melhor hipótese dado um contexto. Mesmo que este tipo de inferência não tenha a certeza da inferência dedutiva, cria-se conclusões que nos fazem considerar a hipótese como plausível e/ou verdadeira.

- Vantagem:

Dos três tipos de inferência esta é a que mais pode acrescentar informação, caso a inferência seja condizente com a realidade e possa se sustentar com argumentos. Quando usada adequadamente, pode ser bastante útil informações a priori (priors) em estatística Bayesiana.

- Desvantagem:

Entretanto, é mais fraca do ponto de vista lógico, no sentido de não haver certeza alguma no que se está concluindo, havendo apenas formação de hipóteses que podem sustentar a análise.

- Exemplo:

O pesquisador está pela primeira vez trabalhando big data que requer uso intenso de GPU, horas de processamento, milhões de amostras, milhões de parâmetros ajustáveis, etc. Ele está treinando uma arquitetura do T5 da Google para traduzir sentenças do inglês para português. Ao iniciar o treinamento percebe que o tempo de processamento de uma época está na ordem de 10 horas. Como o pesquisador não dispõe de tempo para treinar por algumas épocas, decide que precisa diminuir o tempo de treinamento, mas tentando perder o mínimo de qualidade possível. Assim, o pesquisador decide tentar usar uma precisão de 16-bits ao invés da tradicional 32-bits. Essa foi uma escolha dentre inúmeras outras (abaixar batch size, aumentar learning rate, diminuir tamanho das sequências de entrada/saída, usar um modelo menos parâmetros, alterar o scheduling factor, etc). Dentre todas as ações a se tomar, o pesquisador toma como primeiro palpite a tentativa de usar outro valor de precisão, com o pensamento de que este pode oferecer o melhor tradeoff entre custo computacional, mantendo qualidade na tradução de seu modelo.

Por quê treinar uma rede neural está associado a um processo de inferência indutiva?

No treinamento de uma rede neural, um dos principais conceitos está no aprendizado através de dados. O que procura-se no treinamento, é a distribuição original e desconhecida que gerou os dados, para que se possa generalizar para além dos dados que se dispõe. Assim, se pudermos modelar uma distribuição, baseado nos dados de treinamento, podemos tirar conclusões a respeito de qualquer dado possível que esteja submetido àquela distribuição (se a generalização estiver boa).

Uma das definições de aprendizado de máquina, dada por Tom Mitchell (1997) é: "diz-se

que um programa de computador aprendeu de uma experiência  $E$  com respeito a uma certa tarefa  $T$  e com certa performance  $P$ , se sua performance em  $T$  medida por  $P$  aumenta com a experiência  $E$ ". Nessa afirmação, pode-se ver claramente os princípios de inferência indutiva, havendo a questão da experiência em  $E$  e capacidade de generalização em  $T$  medida por  $P$ .

Por fim, temos exatamente a situação em que parte-se de casos particulares (dados de treinamento) a fim de se inferir o caso geral (distribuição que gerou os dados / capacidade de generalização).

## Question 6

### Artigo 1:

Obs.: This question will be answered in Portuguese.

O primeiro paper selecionado é de 2002 com 2173 citações. Esse paper foi publicado no JAMA (The Journal of the American Medical Association) que é um jornal da área médica com 48 publicações por ano pela AMA (American Medical Association).

- **title:** Effects of Cognitive Training Interventions With Older Adults - A Randomized Controlled Trial
- **year:** 2002
- **cited by:** 2173
- **publication:** JAMA
- **reference:** Ball K, Berch DB, Helmers KF, et al. Effects of Cognitive Training Interventions With Older Adults: A Randomized Controlled Trial. JAMA. 2002;288(18):2271–2281. doi:10.1001/jama.288.18.2271

Neste artigo foram avaliadas três intervenções de treinamento cognitivo em pessoas entre 65 e 94 anos, totalizando 2832 voluntários. Os participantes foram aleatoriamente divididos em quatro grupos:

- treinamento para memória: memória diárias verbais (711 pessoas)
- treinamento de raciocínio: habilidade em resolver problemas que seguem padrões (705 pessoas)
- velocidade de processamento: identificação e busca visual (712 pessoas)
- grupo de controle: grupo sem contato apenas para comparação(704 pessoas).

As intervenções melhoraram as habilidades cognitivas ao qual o voluntário foi submetido por até dois anos. 87% dos participantes do grupo de velocidade de processamento, 74% do grupo de raciocínio e 26% dos participantes do treinamento de memória apresentaram melhoras cognitivas confiáveis imediatamente após o período de intervenção. Por fim, os autores concluem que os resultados corroboram com a efetividade e durabilidade de intervenções de treinamento cognitivos em idosos para o desenvolvimento de habilidades cognitivas específicas.

## Artigo 2:

O segundo paper selecionado é de 2009 e conta com 315 citações. Esse paper foi publicado no jornal acadêmico *Alzheimer's & Dementia*, que conta com publicações mensais da associação sem fins lucrativos *Journal of the Alzheimer's Association*.

- **title:** Immediate and delayed effects of cognitive interventions in healthy elderly: A review of current literature and future directions
- **year:** 2009
- **cited by:** 315
- **publication:** *Alzheimer's & Dementia* (Volume 5, Issue 1, January 2009, Pages 50-60)
- **reference:** Papp K V, Walsh S J, Snyder P J. Immediate and delayed effects of cognitive interventions in healthy elderly: a review of current literature and future directions. *Alzheimer's and Dementia* 2009; 5(1): 50-60. [PubMed]

Este artigo se enquadra na categoria “review article”, onde os autores revisam a literatura e avaliam os efeitos imediatos e de longo prazo de intervenções cognitivas na saúde de idosos. Por ser publicado em uma revista sobre Alzheimer, este acaba sendo o foco principal do quesito “saúde”.

A metodologia adotada neste artigo foi a busca de diferentes termos relacionados a treinamento cognitivo (ex: brain training, cognitive intervention) em diferentes fontes, sendo que somente estudos publicados como RTCs (randomized controlled trials) foram incluídos nas análises.

Os tipos de treinamento cognitivos estudados são os mesmos do artigo citado anteriormente: memória, raciocínio e velocidade de processamento. Adicionalmente, foi incluído também um treinamento multimodal, que combina mais de uma categoria ou envolve dois ou mais sistemas sensoriais.

Os resultados mostram que a literatura neste tema é limitada, que não há acompanhamento por tempo suficiente após os treinamentos e que há falta de consenso em quais seriam os treinamentos cognitivos mais eficientes.

Por fim, os autores reconhecem que os treinamentos melhoram habilidades específicas dos participantes, mas que isso não necessariamente generaliza bem para outras atividades. Como exemplo, eles citam que os participantes que receberam treinamento para memória, conseguiram sim melhorias neste quesito, mas obtiveram performance baixa em raciocínio. O mesmo vale no sentido contrário: participantes que receberam treinamento para raciocínio, mostraram melhoras neste quesito, mas sem efeitos em outras tarefas, como memória. Assim, os autores concluem que os treinamentos cognitivos ajudam no desenvolvimento da habilidade da tarefa na qual a pessoa foi submetida, mas que em se tratando de prevenção ao Alzheimer, que não foram encontradas evidências de que programas de intervenções cognitivas estruturadas atrasam ou diminuem a progressão do Alzheimer em idosos.