# IA353 – Redes Neurais (1s2021)

## Exercícios Conceituais 1 – EC 1 - Atividade Individual – Peso 5

## Data de entrega da resolução (por e-mail): 04/05/2021

**Questões associadas a outros oferecimentos da disciplina, com gabarito. Servem de preâmbulo para as atividades associadas a este EC 1.**

**Questão Resolvida 1)**
Tomando a definição de *generalized Tikhonov regularization* ( https://en.wikipedia.org/wiki/Tikhonov_regularization ):

$$\|A\mathbf{x}-\mathbf{b}\|_P^2+\|\mathbf{x}-\mathbf{x}_0\|_Q^2$$

deduza passo-a-passo a solução ótima $\mathbf{x}^*$ apresentada, para quaisquer matrizes simétricas *P* e *Q*. Procure justificar o emprego de um coeficiente de regularização fixo e igual a 1 na formulação acima.

Resolução:

$$J(\mathbf{x})=\|A\mathbf{x}-\mathbf{b}\|_P^2+\|\mathbf{x}-\mathbf{x}_0\|_Q^2=(A\mathbf{x}-\mathbf{b})^T P(A\mathbf{x}-\mathbf{b})+(\mathbf{x}-\mathbf{x}_0)^T Q(\mathbf{x}-\mathbf{x}_0)$$

$$=(\mathbf{x}^T A^T-\mathbf{b}^T)(PA\mathbf{x}-P\mathbf{b})+(\mathbf{x}^T-\mathbf{x}_0^T)(Q\mathbf{x}-Q\mathbf{x}_0)$$

$$=\mathbf{x}^T A^T PA\mathbf{x}-\mathbf{x}^T A^T P\mathbf{b}-\mathbf{b}^T PA\mathbf{x}+\mathbf{b}^T P\mathbf{b}+\mathbf{x}^T Q\mathbf{x}-\mathbf{x}^T Q\mathbf{x}_0-\mathbf{x}_0^T Q\mathbf{x}+\mathbf{x}_0^T Q\mathbf{x}_0$$

$$\frac{\partial J(\mathbf{x})}{\partial \mathbf{x}}=2A^T PA\mathbf{x}-A^T P\mathbf{b}-A^T P\mathbf{b}+2Q\mathbf{x}-Q\mathbf{x}_0-Q\mathbf{x}_0=2A^T PA\mathbf{x}-2A^T P\mathbf{b}+2Q\mathbf{x}-2Q\mathbf{x}_0$$

Aplicando a condição necessária de otimalidade, resulta:

$$\left.\frac{\partial J(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}^*}=0\Rightarrow A^T PA\mathbf{x}^*-A^T P\mathbf{b}+Q\mathbf{x}^*-Q\mathbf{x}_0=0$$

$$(A^T PA+Q)\mathbf{x}^*=A^T P\mathbf{b}+Q\mathbf{x}_0$$

As matrizes *P* e *Q* são sabidas serem simétricas, mas não são de nosso conhecimento. No entanto, a resposta desejada supõe a existência da inversa de $A^T PA+Q$, o que passaremos a considerar a partir daqui. Assim, multiplicando à esquerda por $(A^T PA+Q)^{-1}$ nos dois lados da equação acima, obtém-se:

$$(A^T PA+Q)^{-1}(A^T PA+Q)\mathbf{x}^*=(A^T PA+Q)^{-1}(A^T P\mathbf{b}+Q\mathbf{x}_0)$$

$$\mathbf{x}^*=(A^T PA+Q)^{-1}(A^T P\mathbf{b}+Q\mathbf{x}_0).$$

O coeficiente de regularização é fixo e igual a 1 porque a matriz *Q* já o inclui. Isso pode ser constatado de duas formas alternativas:

- Considere um coeficiente de regularização diferente de 1:

$$\|A\mathbf{x}-\mathbf{b}\|_P^2+\lambda\|\mathbf{x}-\mathbf{x}_0\|_Q^2$$

Essa nova formulação vai levar à solução:

$$\mathbf{x}^*=\left(A^TPA+\lambda Q\right)^{-1}\left(A^TP\mathbf{b}+\lambda Q\mathbf{x}_0\right)$$

Fazendo $Q'=\lambda Q$, retorna-se à formulação com coeficiente de regularização igual a 1:

$$\|A\mathbf{x}-\mathbf{b}\|_P^2+\|\mathbf{x}-\mathbf{x}_0\|_{Q'}^2$$

- Tomando $P=I$, $Q=\lambda I$ e $\mathbf{x}_0=0$, obtém-se a formulação original que estávamos acostumados a tratar, o que indica que o coeficiente de regularização de fato está embutido na matriz *Q*:

$$\|A\mathbf{x}-\mathbf{b}\|^2+\lambda\|\mathbf{x}\|^2.$$

**Questão Resolvida 2)**

(a) Prove que matrizes $n\times n$ que podem ser decompostas na forma $M=N^TN$ com $N\in\Re^{m\times n}$, são matrizes simétricas e semi-definidas positivas.

Resolução:

$M^T=\left(N^TN\right)^T=N^T\left(N^T\right)^T=N^TN=M$. Logo, *M* é uma matriz simétrica.

Por definição, uma matriz *M* de dimensão $n\times n$ é semi-definida positiva se $\mathbf{x}^TM\mathbf{x}\geq 0,\ \forall\mathbf{x}\in\Re^n$. Usando a decomposição, resulta:

$$\mathbf{x}^TM\mathbf{x}=\mathbf{x}^TN^TN\mathbf{x}=\left(N\mathbf{x}\right)^T\left(N\mathbf{x}\right)=\|N\mathbf{x}\|_2^2.$$

Por definição, a norma euclidiana é tal que $\|\mathbf{z}\|_2\geq 0,\ \forall\mathbf{z}\in\Re^m$, ou seja:

$$\mathbf{x}^TM\mathbf{x}=\|N\mathbf{x}\|_2^2\geq 0,\ \forall\mathbf{x}\in\Re^n,$$

demonstrando assim que *M* é semi-definida positiva.

(b) Supondo *m* > *n* e tomando *N* de posto completo, prove que *M* tem que ser definida positiva.

Por definição, uma matriz *M* de dimensão $n\times n$ é definida positiva se $\mathbf{x}^TM\mathbf{x}>0,\ \forall\mathbf{x}\in\Re_+^n$, onde $\Re_+^n$ representa o espaço $\Re^n$ sem o vetor nulo. Se a matriz *N* tem posto completo, então o seu espaço nulo tem dimensão zero, o que indica que somente o vetor nulo é solução de $N\mathbf{x}=0$. Você pode recorrer também à definição de independência linear entre vetores e aplicar a $N\mathbf{x}=0$, onde $N\mathbf{x}$ nada mais é que uma combinação linear das colunas de *N*. Logo:

$\mathbf{x}^T M\mathbf{x} = \|N\mathbf{x}\|_2^2 > 0, \forall \mathbf{x} \in \Re_+^n$,

demonstrando assim que *M* é definida positiva.

(c) Usando o conceito de matriz ortogonal, prove que *N* pode não ser única, para um mesmo *M*.

Resolução:

Por definição, uma matriz ortogonal *Q* é tal que $Q^T Q = I$, ou seja, $Q^T = Q^{-1}$. Sendo assim, tomando uma matriz *Q* ortogonal qualquer e de dimensão apropriada, sabe-se que $QN \neq N$. No entanto, se fizermos:

$(QN)^T (QN) = N^T Q^T QN = N^T IN = N^T N = M$. Logo, existe uma matriz diferente de *N* que produz *M*.

(d) Apresente uma proposta para obter *N*, dada uma matriz *M* definida positiva.

Resolução:

Sendo *M* uma matriz simétrica, é sempre possível obter *n* autovetores ortogonais para *M*, que irão compor as colunas de uma matriz *T*. Com essa matriz *T*, é possível chegar a uma matriz diagonal $\Lambda$ que tenha os autovalores de *M* em sua diagonal, na forma:

$M = T\Lambda T^T$.

Como *M* é definida positiva, seus autovalores são todos positivos. Sendo assim, é possível obter uma matriz diagonal $\Lambda_1$ tal que os elementos da diagonal são a raiz quadrada dos autovalores de *M*. Sendo assim, obtém-se:

$$M = T\Lambda T^T = T\Lambda_1\Lambda_1 T^T = T\Lambda_1^T\Lambda_1 T^T = \left(\Lambda_1 T^T\right)^T \left(\Lambda_1 T^T\right).$$

Agora é só fazer $N = \Lambda_1 T^T$, mostrando que é possível obter *N* a partir da decomposição espectral de *M*.

Uma alternativa para resolver esta questão seria recorrer à Fatoração de Cholesky ( https://en.wikipedia.org/wiki/Cholesky_decomposition ), mas aí teria que mostrar como fica a matriz *N*.

**Questão Resolvida 3)**

Uma pessoa recebeu um modelo linearizado $f_L(x, y)$ para a função $f(x, y) = x\sqrt{y}$, em torno de um determinado ponto $(x_0, y_0)$. Essa pessoa agora precisa utilizar a versão linear $f_L(x, y)$, mas não é capaz de se lembrar de todos os coeficientes de $f_L(x, y)$, a qual tem a forma:

$$f_L(x, y) = 2x + py - 8,$$

onde o coeficiente *p* é uma incógnita. Além disso, a pessoa não se lembra do ponto $(x_0, y_0)$ em torno do qual foi feita a linearização. Mostre a esta pessoa como obter os referidos valores de $(x_0, y_0)$ e *p*, apresentando os passos necessários e os respectivos

valores numéricos. Nota: Expansão de uma função $f$: $\Re^n \rightarrow \Re$ em série de Taylor em torno do ponto $\mathbf{x}^* \in \Re^n$:

$$f(\mathbf{x}) = f(\mathbf{x}^*) + \nabla f(\mathbf{x}^*)^T (\mathbf{x} - \mathbf{x}^*) + \frac{1}{2}(\mathbf{x} - \mathbf{x}^*)^T \nabla^2 f(\mathbf{x}^*)(\mathbf{x} - \mathbf{x}^*) + O(3).$$

Resolução:

Temos que lembrar que a linearização é sempre realizada em algum ponto específico, da mesma forma que o vetor gradiente (para o treinamento supervisionado de redes neurais) sempre é obtido em um ponto específico da superfície de erro. Com isso, a obtenção de $f_L(x,y)$, que é a versão linear de $f(x,y) = x\sqrt{y}$, requer expandir $f(x,y)$ em série de Taylor até primeira ordem, em torno de um determinado ponto $(x_0, y_0)$. Procedendo dessa forma, no entorno de $(x_0, y_0)$, resulta:

$$f(x,y) \cong f_L(x,y) = f(x_0,y_0) + \left[\frac{\partial f(x,y)}{\partial x} \quad \frac{\partial f(x,y)}{\partial y}\right]_{(x,y)=(x_0,y_0)} \begin{bmatrix} x - x_0 \\ y - y_0 \end{bmatrix}$$

$$= x_0\sqrt{y_0} + \left[\sqrt{y} \quad \frac{x}{2\sqrt{y}}\right]_{(x,y)=(x_0,y_0)} \begin{bmatrix} x - x_0 \\ y - y_0 \end{bmatrix}$$

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

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

Logo:

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

o que conduz a:

$$\sqrt{y_0} = 2 \Rightarrow y_0 = 4$$

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

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

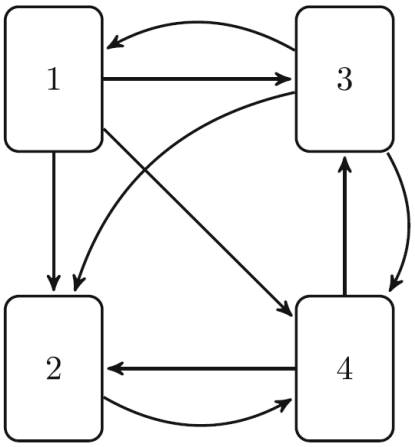
Cabe lembrar que é sempre possível conferir se o resultado está correto, ao comparar $f(x,y)$ com $f_L(x,y)$ no ponto $(x_0, y_0)$.

A seguir, se iniciam as atividades do EC 1, com questões conceituais a serem resolvidas.

**Questão 1) (2,0 pontos)**

O algoritmo *PageRank* é um método para avaliar a "importância" de documentos com *links* mútuos, como páginas da Web, com base na estrutura de *links*. Foi desenvolvido por Sergei Brin e Larry Page, os fundadores do Google Inc., na Universidade de Stanford no final dos anos 1990. A ideia básica do algoritmo é a seguinte: Ao invés de contar *links*, o *PageRank* essencialmente interpreta um link da página *A* para a página *B* como um voto da página *A* para a página *B*. O *PageRank* avalia a importância de uma página pelo número de votos recebidos, mas também considerando a importância da página que lança o voto, uma vez que os votos de algumas páginas têm um valor maior e, portanto, também atribuem um valor maior à página para a qual apontam. As páginas importantes terão uma classificação mais elevada e, portanto, levarão a uma posição superior nos resultados da pesquisa.

Vamos descrever essa ideia matematicamente. Para um determinado conjunto de páginas da Web, a cada página *k* será atribuído um valor de importância $x_k \geq 0$. Uma página *k* é mais importante do que uma página *j* se $x_k > x_j$. Se uma página *k* tem um *link* para uma página *j*, dizemos que a página *j* tem um *backlink* da página *k*. Na descrição acima, esses *backlinks* são os votos. Por exemplo, considere a seguinte estrutura de links:




Aqui, a página 1 contém *links* para as páginas 2, 3 e 4 e um *backlink* da página 3. A abordagem mais fácil para definir a importância das páginas da web é contar seus *backlinks*; quanto mais votos são lançados para uma página, mais importante ela é. Em nosso exemplo, isso dá os valores de importância:

$$x_1 = 1; x_2 = 3; x_3 = 2; x_4 = 3.$$

As páginas 2 e 4 são, portanto, as páginas mais importantes, empatando no *ranking*. No entanto, a descrição do parágrafo inicial do enunciado desta questão sugere que os *backlinks* de páginas importantes são mais importantes para o valor de uma página do que os de páginas menos importantes. Essa ideia pode ser modelada definindo $x_k$ como a soma de todos os valores de importância dos *backlinks* da página *k*. Em nosso exemplo, isso resulta em quatro equações que devem ser satisfeitas simultaneamente:

$$x_1 = x_3; x_2 = x_1 + x_3 + x_4; x_3 = x_1 + x_4; x_4 = x_1 + x_2 + x_3.$$

Uma desvantagem dessa abordagem é que ela não leva em consideração o número de *links* das páginas. Assim, seria possível aumentar (significativamente) a importância

de uma página apenas adicionando *links* a essa página. Para evitar isso, os valores de importância dos *backlinks* no algoritmo *PageRank* são divididos pelo número de *links* da página correspondente. Isso cria uma espécie de "democracia na internet": cada página pode votar parcialmente em múltiplas páginas, sendo que o total de votos parciais de cada página soma um voto. Em nosso exemplo, isso conduz às equações:

$$x_1 = \frac{x_3}{3};\ x_2 = \frac{x_1}{3} + \frac{x_3}{3} + \frac{x_4}{2};\ x_3 = \frac{x_1}{3} + \frac{x_4}{2};\ x_4 = \frac{x_1}{3} + x_2 + \frac{x_3}{3}.$$

Existem, portanto, quatro equações para quatro incógnitas e todas as equações são lineares.

1. Monte o sistema linear de equações, na forma $A\mathbf{x} = \mathbf{x}$ para as quatro equações acima. Observação: Analisar e resolver de forma eficiente este sistema, considerando dimensões da ordem do número de páginas da Web, é uma das tarefas mais importantes da Álgebra Linear.
2. Tomando agora uma matriz genérica, $A \in \Re^{n\times n}$, prove que os autovalores da matriz $A^T$ são iguais aos autovalores da matriz *A*. Sugestão: Use a propriedade de determinante de uma matriz transposta.
3. Por construção, é sabido que $a_{ij} \geq 0$ e $\sum_{i=1}^{n} a_{ij} = 1$ para $j = 1,\cdots,n$. Matrizes com esta propriedade são chamadas *column-stochastic*. Com base nesse aspecto construtivo e na demonstração do item (2), mostre que a matriz *A* sempre vai ter um autovalor igual a 1.
4. Conseguimos, assim, converter o problema de avaliar a importância das páginas da Web em um problema de obtenção do autovetor correspondente ao autovalor unitário da matriz *A*. Com a demonstração do item (3), fica garantido que o problema do *PageRank* sempre tem solução, sendo o autovetor correspondente ao autovalor unitário, o qual sempre existe. No entanto, a ideia de *ranking* requer que todos os elementos deste autovetor sejam não-negativos. Prove que, se a matriz $A \in \Re^{n\times n}$, além de ser *column-stochastic*, tem todos os seus elementos positivos, então ou **x** ou –**x** tem todos os seus elementos positivos. Sugestão: Lembre-se que, como **x** é o autovetor correspondente ao autovalor unitário, então vale a expressão:

   $$x_i = \sum_{j=1}^{n} a_{ij} x_j,\ i = 1,\cdots,n.$$

   Suponha que **x** ou –**x** não tem todos os seus elementos positivos e use a expressão acima para chegar a um absurdo.
5. Com todas essas condições e propriedades acumuladas, é possível demonstrar que o problema de PageRank tem solução única, ou seja, para $A \in \Re^{n\times n}$ com todos os seus elementos positivos e *column-stochastic*, existe um único autovetor com elementos positivos tal que $A\mathbf{x} = \mathbf{x}$, sendo sempre possível fazer $\sum_{i=1}^{n} x_i = 1$. Este autovetor é chamado autovetor *Perron*. No entanto, na prática, a matriz *A* vai conter muitos elementos nulos. Para garantir a existência de solução, foi concebido um truque algébrico usando uma matriz $S \in \Re^{n\times n}$ tal que $s_{ij} = 1/n,\ i, j = 1,\cdots,n$. Obviamente *S* tem todos os seus elementos positivos e é *column-stochastic*. Fazendo agora:

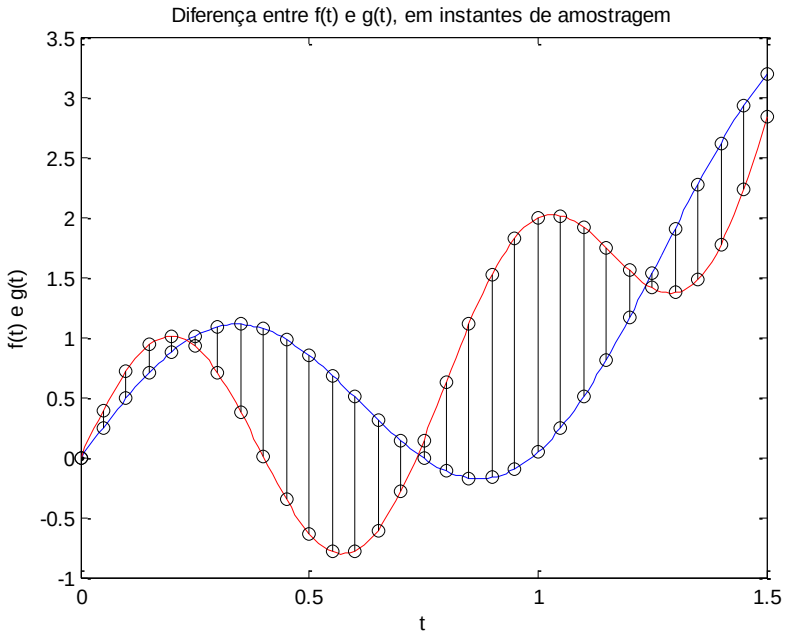
   $$\hat{A}(\alpha) = (1-\alpha)A + \alpha S,\ \alpha \in (0,1],$$

prove que $\hat{A}(\alpha)$ tem todos os seus elementos positivos e é *column-stochastic*. Em seguida, mostre que $\hat{A}(\alpha)\mathbf{x} \cong \mathbf{x}$ para valores elevados de *n* e tomando $\sum_{i=1}^{n} x_i = 1$.

Observação: A teoria de autovalores e autovetores de matrizes com elementos positivos (ou não-negativos) é uma área importante da Teoria de Matrizes, uma vez que essas matrizes surgem em muitas aplicações de grande interesse prático. A solução prática do problema de autovalores com a matriz $\hat{A}(\alpha)$ é um tópico da área de Álgebra Linear Numérica, não sendo tratado aqui.

**Questão 2) (1,0 pontos)**

Um engenheiro de controle recebeu a missão de identificar parâmetros de um sistema dinâmico de ordem reduzida, no caso, ordem 1, de modo que sua resposta dinâmica ao degrau mais se aproximasse da resposta ao degrau de um sistema dinâmico de ordem superior, no sentido de quadrados mínimos. O engenheiro sabia que é possível comparar duas funções no tempo obtendo a diferença entre elas em instantes de amostragem (veja a figura abaixo), elevando cada uma dessas diferenças ao quadrado e somando todas elas.




1. Dispondo dos valores da resposta ao degrau ao longo dos instantes de amostragem do sistema dinâmico de ordem superior

$$\{x(t_0), x(t_1), \cdots, x(t_N)\}, \text{ onde } t_k - t_{k-1} = h, k = 1, \cdots, N,$$

e sabendo que a dinâmica de primeira ordem é dada por:

$$q\dot{y}(t) + y(t) = Pu(t), t \in \Re_+,$$

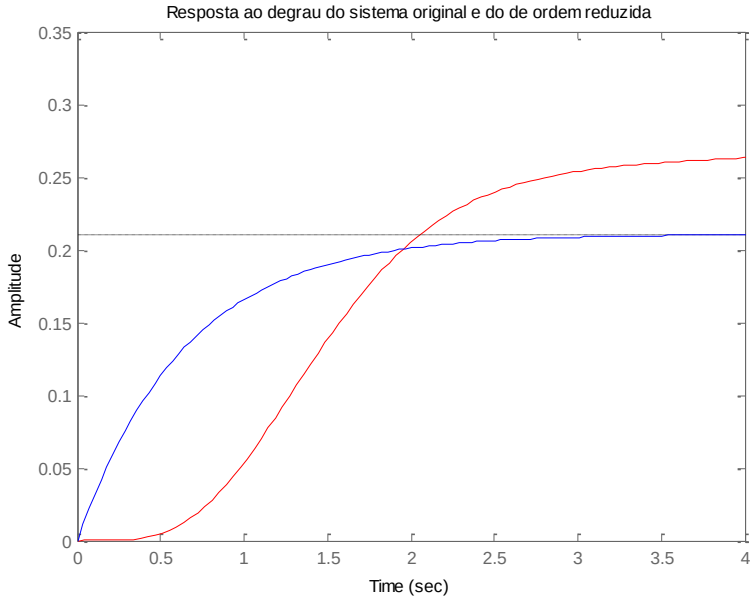
onde *q* e *P* são os parâmetros livres a serem identificados, o engenheiro foi capaz de discretizar esta equação diferencial, usando a aproximação:

$$\dot{y}(t_k) \approx \frac{y(t_{k+1}) - y(t_{k-1})}{2h}.$$

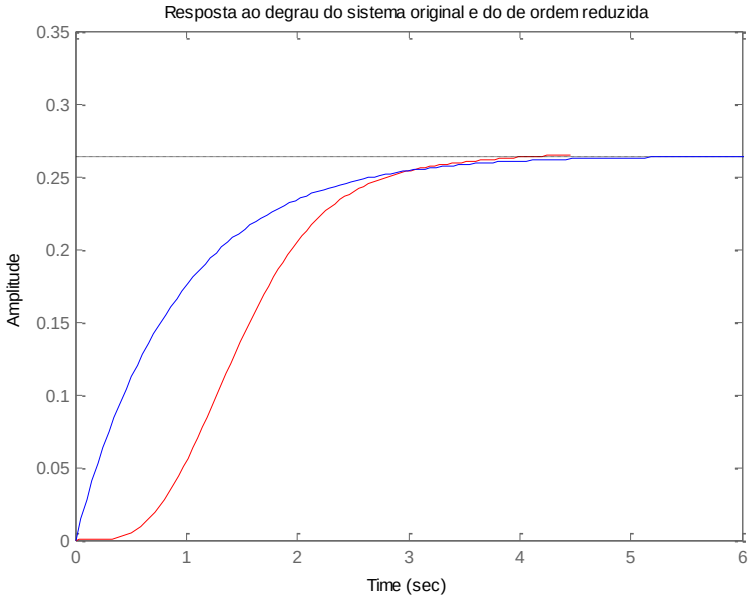
Em seguida, ele montou o sistema linear de equações que precisava ser resolvido para fornecer a solução de quadrados mínimos. Procure então, repetir os passos do engenheiro de controle e forneça o sistema linear de equações, na forma *Ab* = *c*,

sendo *b* o vetor de incógnitas. Uma vez obtido *b* (suponha que você chegou em valores numéricos para os elementos de *b*), o que precisa ser feito para se chegar aos parâmetros do sistema de ordem reduzida (*q* e *P*)?

2. Infelizmente, mesmo realizando todas as etapas corretamente, a solução de quadrados mínimos produziu como resultado:




a qual, embora seja a melhor aproximação de quadrados mínimos para a curva vermelha com um sistema de primeira ordem (cuja resposta ao degrau é a curva azul), não é uma solução aceitável, pois o que é mais relevante ser reproduzido pelo sistema de ordem reduzida, nesta aplicação específica, é a constante de tempo do sistema original (tempo de reação do sistema) e o valor de regime. O engenheiro então realizou uma modificação no problema de quadrados mínimos que permitiu obter o seguinte resultado:



Procure descobrir qual tipo de modificação foi realizada visando o atendimento dos objetivos da tarefa de identificação de sistemas, justificando sua resposta.

**Questão 3) (1,0 pontos)**
Em virtude das conclusões com embasamento experimental levantadas pelo paper de 2018 (https://arxiv.org/pdf/1705.08292.pdf):

**The Marginal Value of Adaptive Gradient Methods in Machine Learning**

**Ashia C. Wilson[♯], Rebecca Roelofs[♯], Mitchell Stern[♯], Nathan Srebro[†], and Benjamin Recht[♯]**
{ashia,roelofs,mitchell}@berkeley.edu, nati@ttic.edu, brecht@berkeley.edu

[♯]University of California, Berkeley
[†]Toyota Technological Institute at Chicago

o(a) aluno(a) é convidado(a) a implementar (Sugestão: se baseie no paper Ruder, S. "An overview of gradient descent optimization algorithms", arXiv:1609.04747v2, 2017) e aplicar o algoritmo ADAM para resolver um sistema linear de equações (nas versões subdeterminada e sobredeterminada), comparando o resultado com aquele produzido por uma proposta básica de gradiente descendente estocástico (SGD, do inglês *Stochastic Gradient Descent*). Note que se trata de um problema convexo, mas que vai ser resolvido de forma iterativa. Seguem propostas de código em Matlab / GNU Octave para o SGD. Apresente o seu código do ADAM e os resultados gráficos, devidamente interpretados.

```
% SGD para o caso de sistema linear sobredeterminado
clear all;
randn('state',0);
N = 10;
Nit = 500;
X = randn(N,2);
S = sign(randn(N,1));
w = (X'*X)\X'*S;
disp('Optimal solution');
disp(w);
w1 = zeros(2,1);
passo = 0.1;
for it=2:Nit,
    w1(:,it) = w1(:,it-1) - (passo/sqrt(it))*(X'*X*w1(:,it-1)-X'*S);
end
figure(1);
title('Stochastic Gradient Descent');
for it = 1:(Nit-1),
    plot([w1(1,it);w1(1,it+1)],[w1(2,it);w1(2,it+1)]);hold on;
    plot(w1(1,it),w1(2,it),'o');
    plot(w1(1,it+1),w1(2,it+1),'o');
end
hold off;
disp('Obtained solution');
disp(w1(:,Nit));
[S X*w X*w1(:,Nit)]
```

```
% SGD para o caso de sistema linear subdeterminado
clear all;
randn('state',0);
N = 10;
Nit = 500;
X = randn(N,2*N);
S = sign(randn(N,1));
w = (X'/(X*X'))*S;
w1 = zeros(2*N,1);
passo = 0.1;
for it=2:Nit,
    w1(:,it) = w1(:,it-1) - (passo/sqrt(it))*(X'*X*w1(:,it-1)-X'*S);
end
figure(1);
title('Stochastic Gradient Descent');
for it = 1:(Nit-1),
    plot([w1(1,it);w1(1,it+1)],[w1(2,it);w1(2,it+1)]);hold on;
    plot(w1(1,it),w1(2,it),'o');
    plot(w1(1,it+1),w1(2,it+1),'o');
end
hold off;
disp('[Minimum Norm Solution  Obtained solution]');
disp([w w1(:,Nit)]);
[S X*w X*w1(:,Nit)]
```

Nota 1: As conclusões extraídas desta atividade conceitual não devem ser estendidas diretamente ao caso do treinamento de uma rede neural profunda, em virtude da grande diferença existente entre os problemas de otimização envolvidos.

Nota 2: Investigue o que ocorre para variadas condições iniciais, não se restringindo à sugestão de partir da origem.

**Questão 4) (1,0 pontos)**
Defina os conceitos de indução, dedução e abdução, recorrendo a textos como em:
https://larspsyll.wordpress.com/2016/01/25/deduction-induction-abduction/
de autoria de Lars Jörgen Pålsson Syll (Doutor em História da Economia).
Explique por que treinar uma rede neural está associado a um processo de inferência indutiva. Em seguida, apresente uma vantagem e uma desvantagem de cada um desses três tipos de inferência (em geral, não restrito ao contexto do curso) e dê um exemplo de cada um no contexto de um pesquisador treinando uma rede neural artificial.
Segue um exemplo ilustrativo desta última atividade da questão:
RACIOCÍNIO INDUTIVO → O erro de treinamento está alto e eu preciso baixar. Nas outras vezes em que eu me defrontei com erros de treinamento altos, bastou aumentar o número de neurônios na rede neural e o erro de treinamento baixou. Logo, a minha decisão é aumentar o número de neurônios.