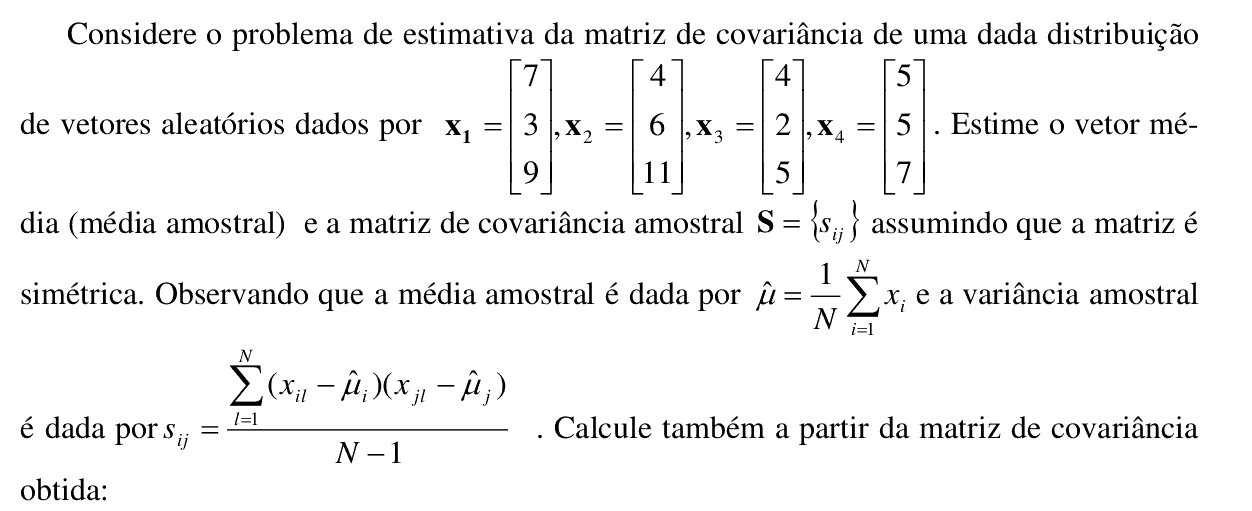
**LISTA 1 - EEC1505: REDES NEURAIS**

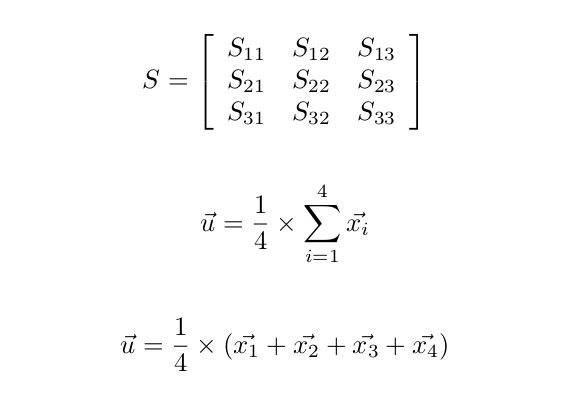
**Aluno:** Pedro Victor Andrade Alves

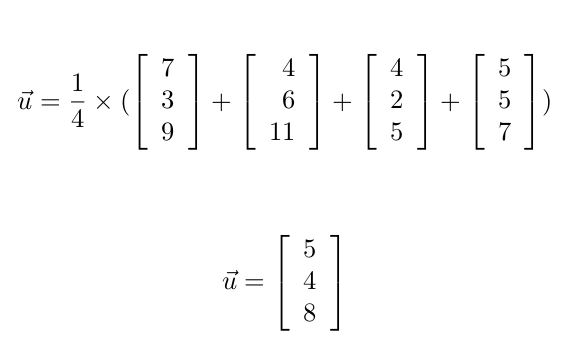
**Matrícula:** 20211022916

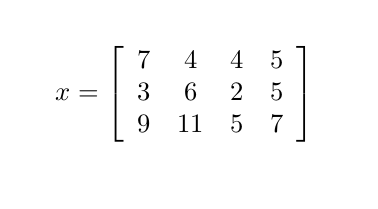
**1º )**

****

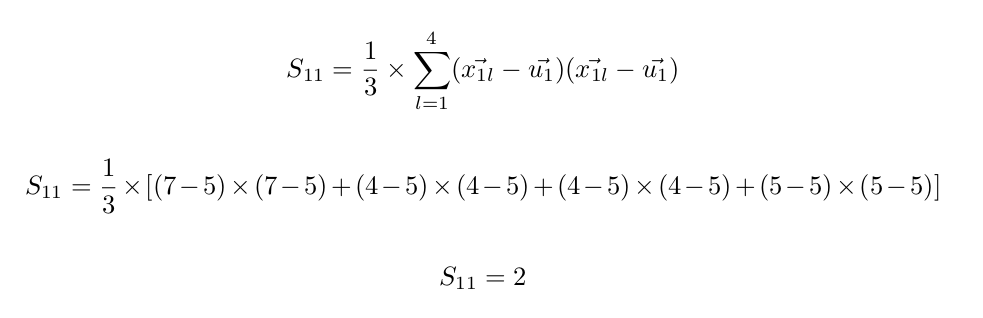
**Resposta:**

****

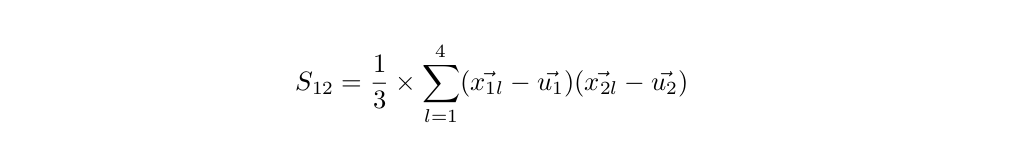
****

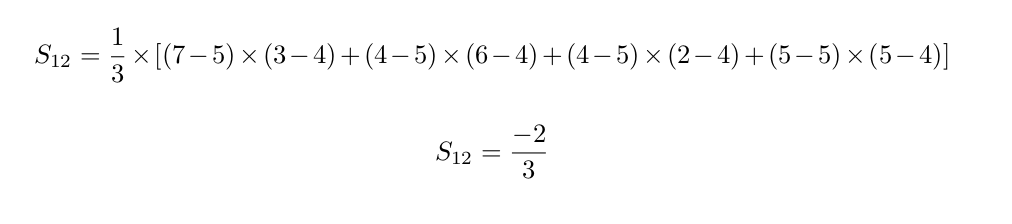
****

**Para S11:**

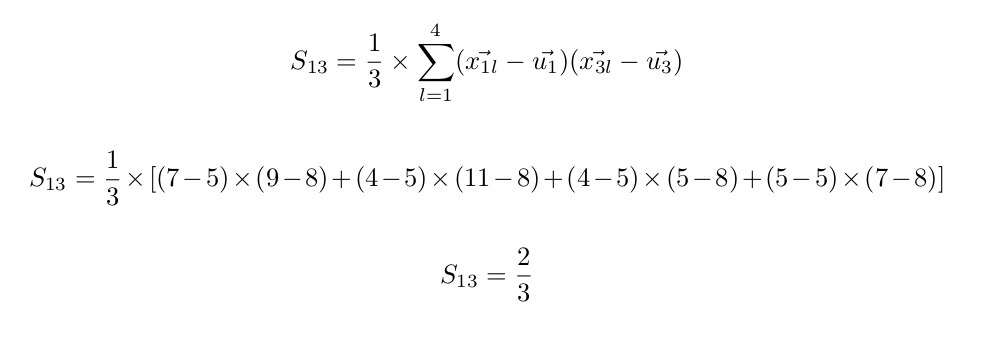
****

**Para S12:**

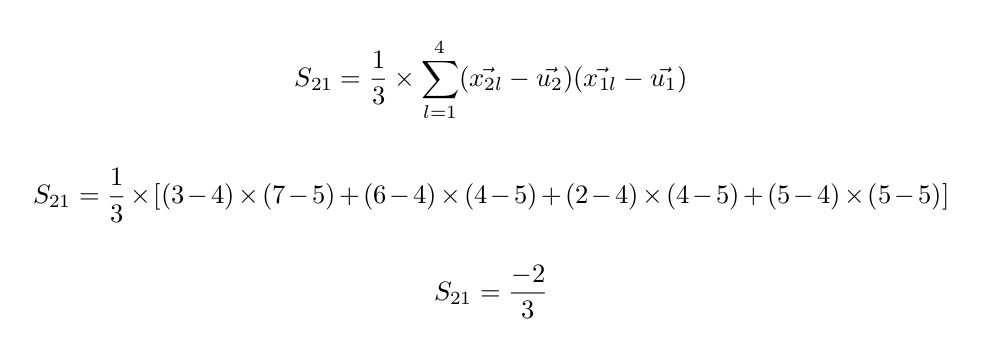
****

****

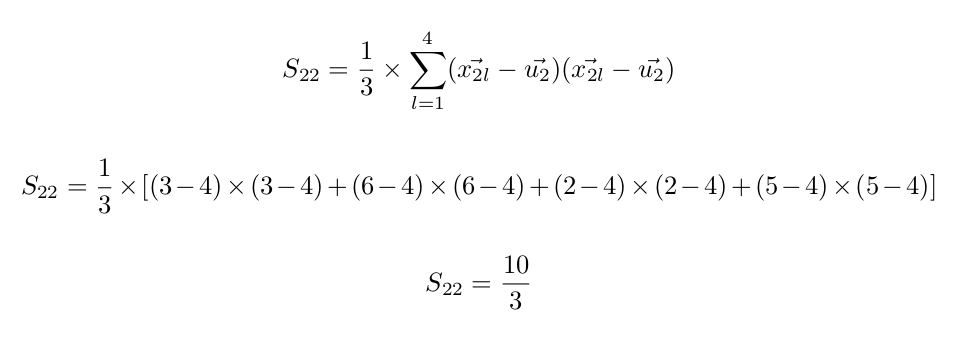
**Para S13:**

****

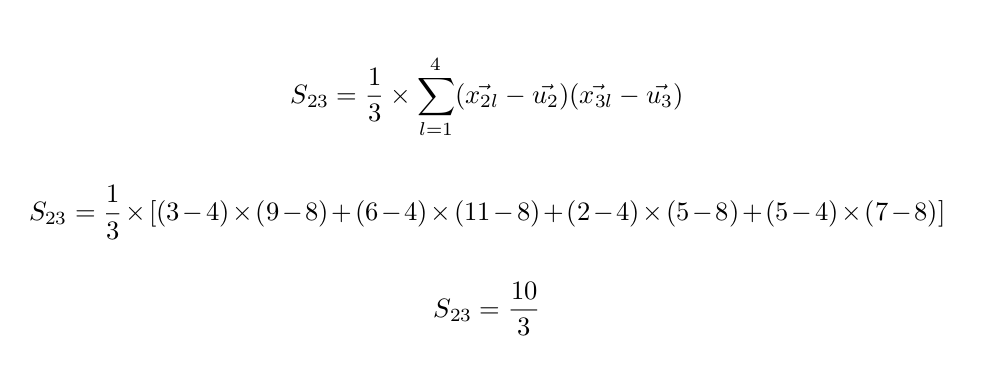
**Para S21:**

****

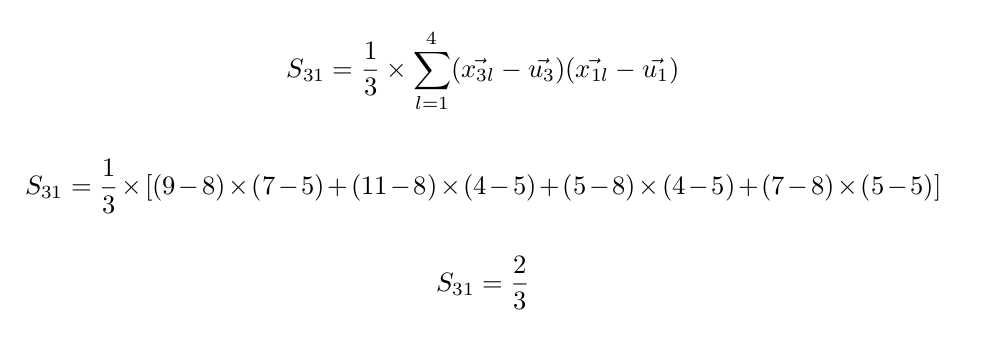
**Para S22:**

****

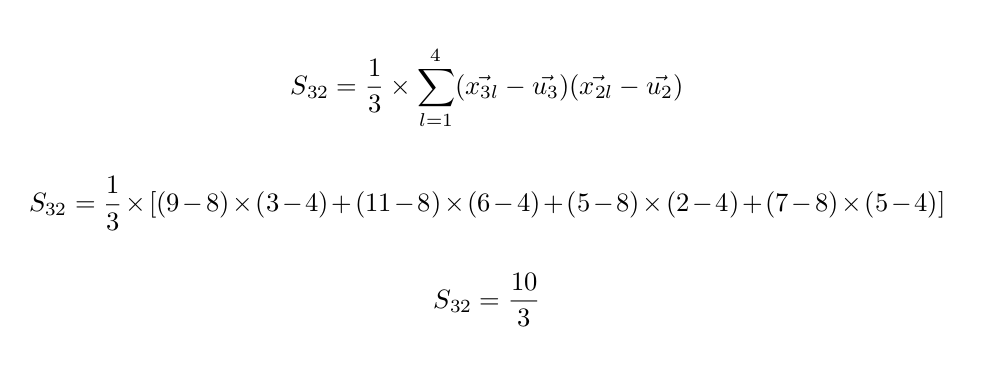
**Para S23:**

****

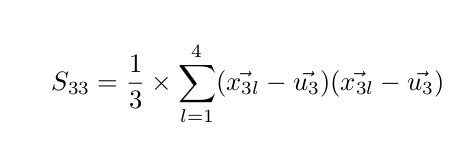
**Para S31:**

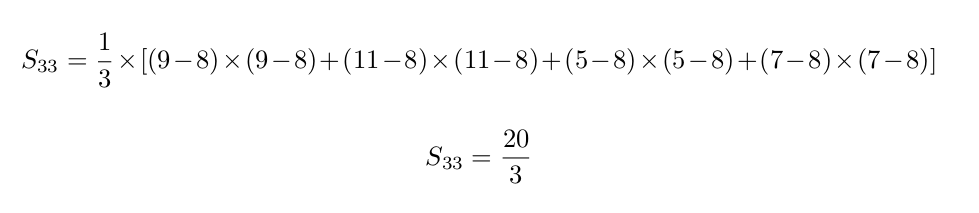
****

**Para S32:**

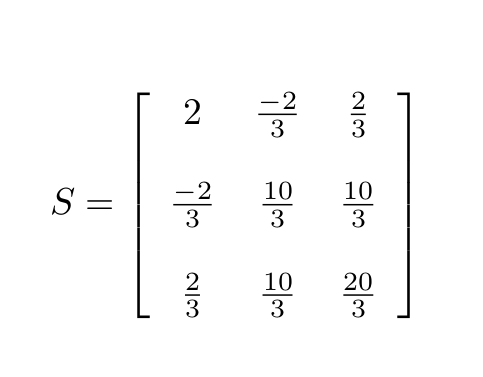
****

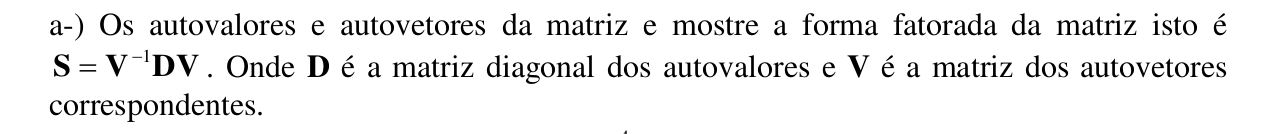
**Para S33:**

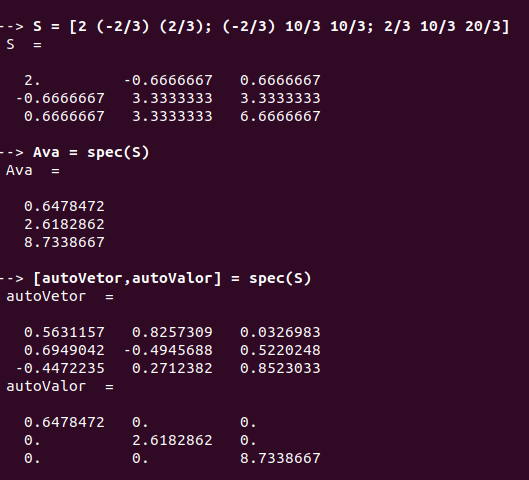
****

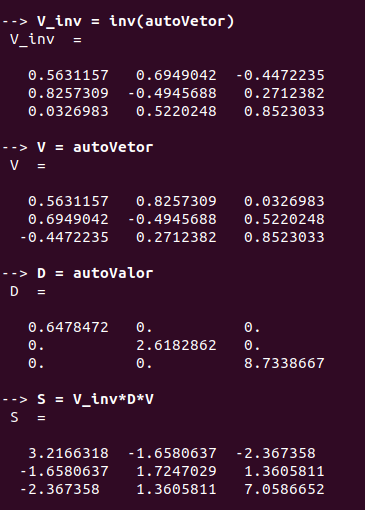
****

**Matriz de covariância:**

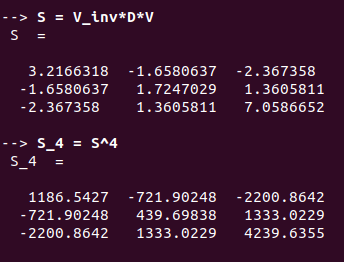
****

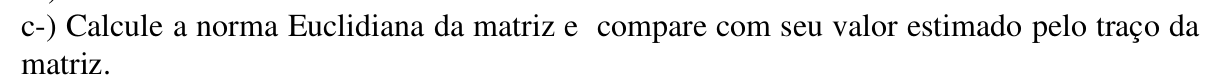
****

****

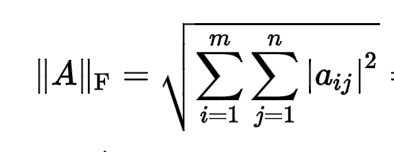
****

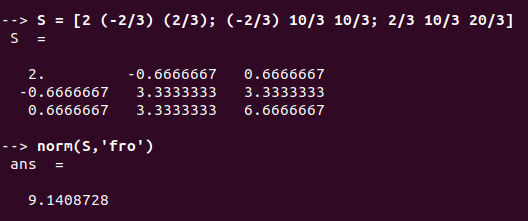
****

****

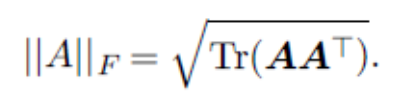
****

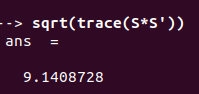
* **Norma de Frobenius (equivalente a norma Euclidiana para vetores)**

****

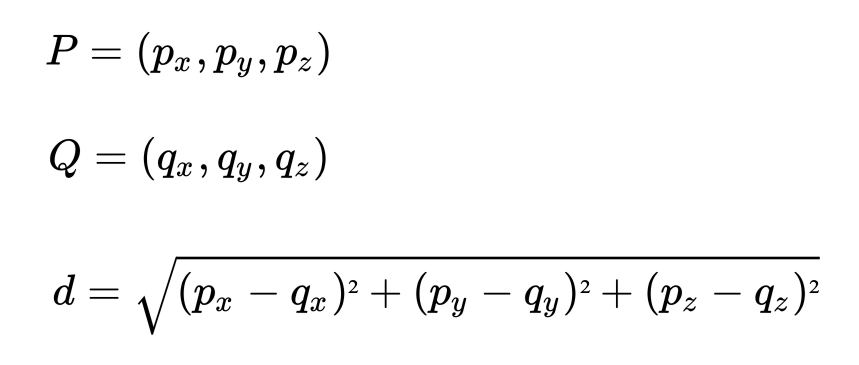
****

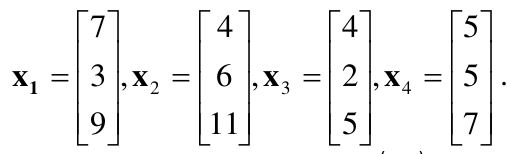
* **Norma de Frobenius pelo traço da matriz**

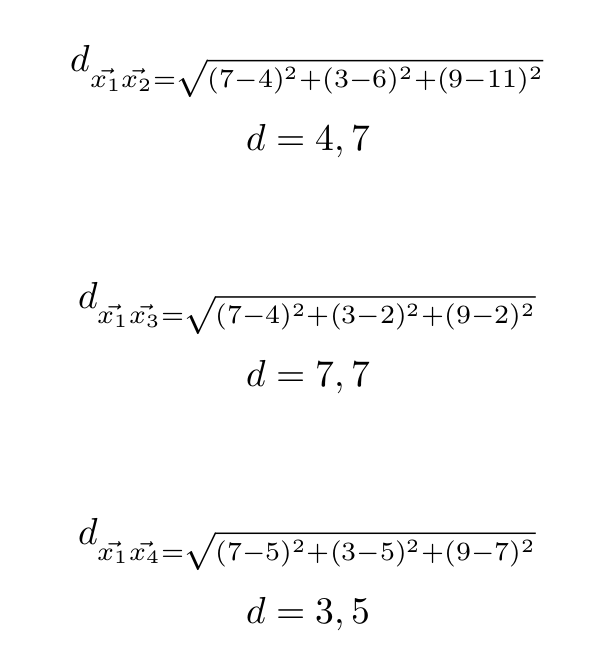
****

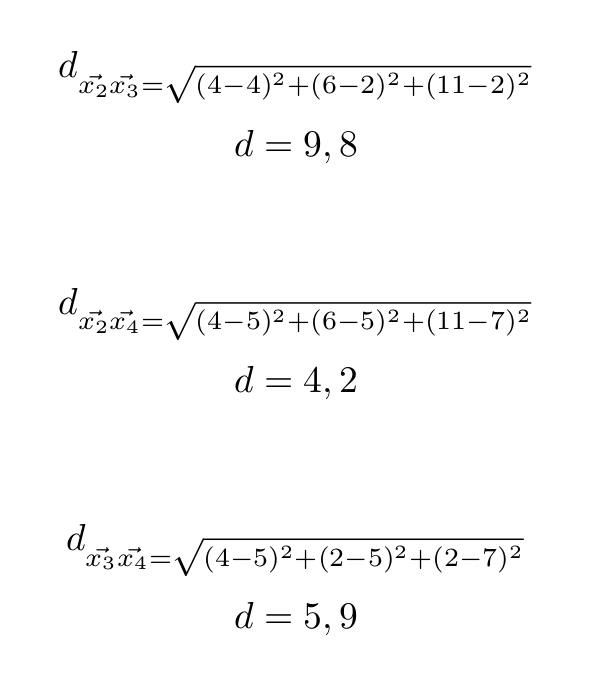
****

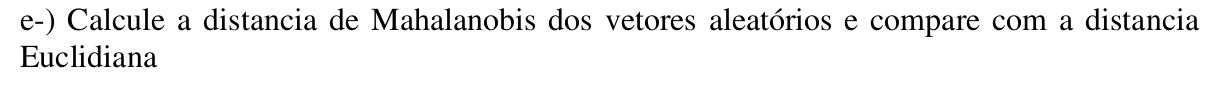
****

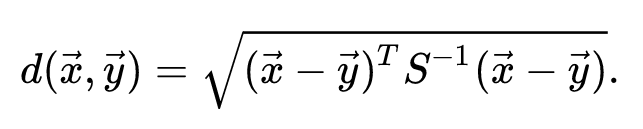
****

****

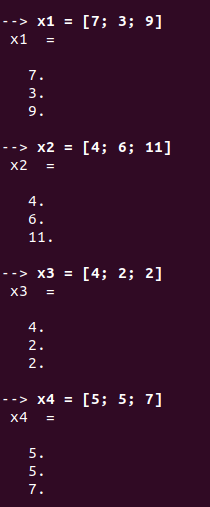
****

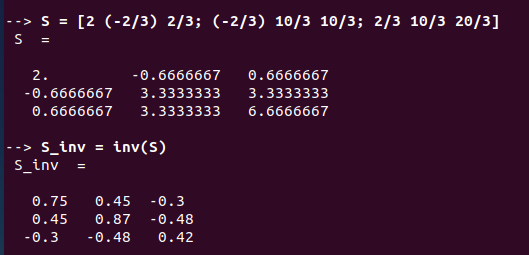
****

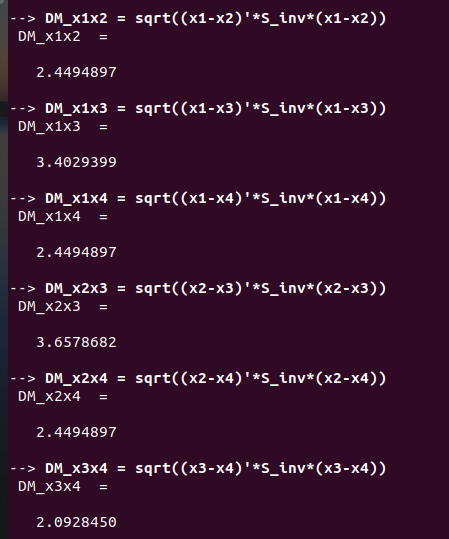
****

****

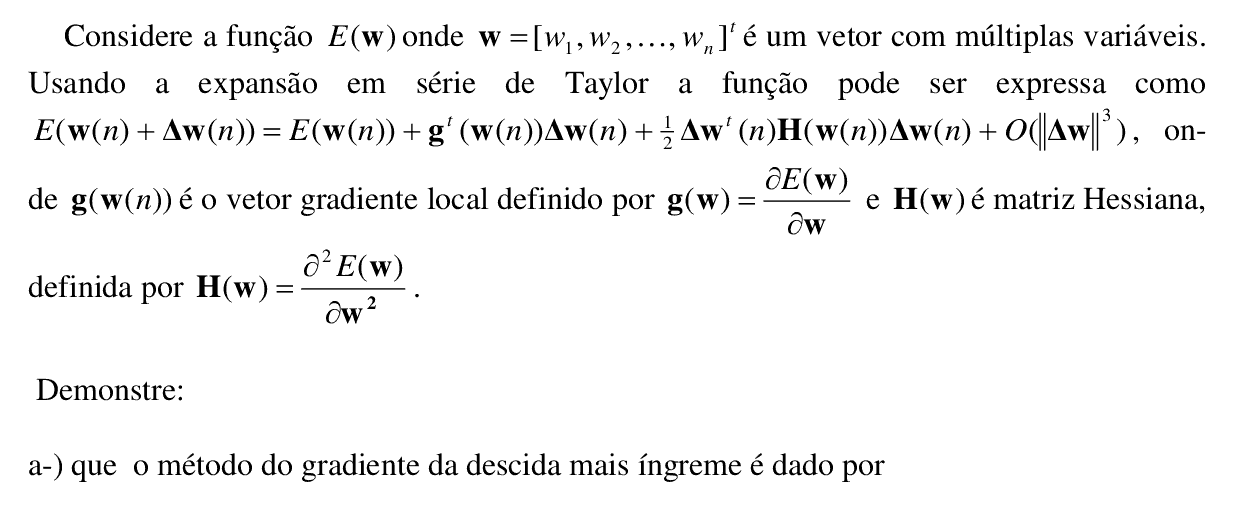
Se diferencia da distância Euclidiana, porque considera as correlações (associações estatísticas entre variáveis) do conjunto de dados e não é influenciada pela escala de medição.

****

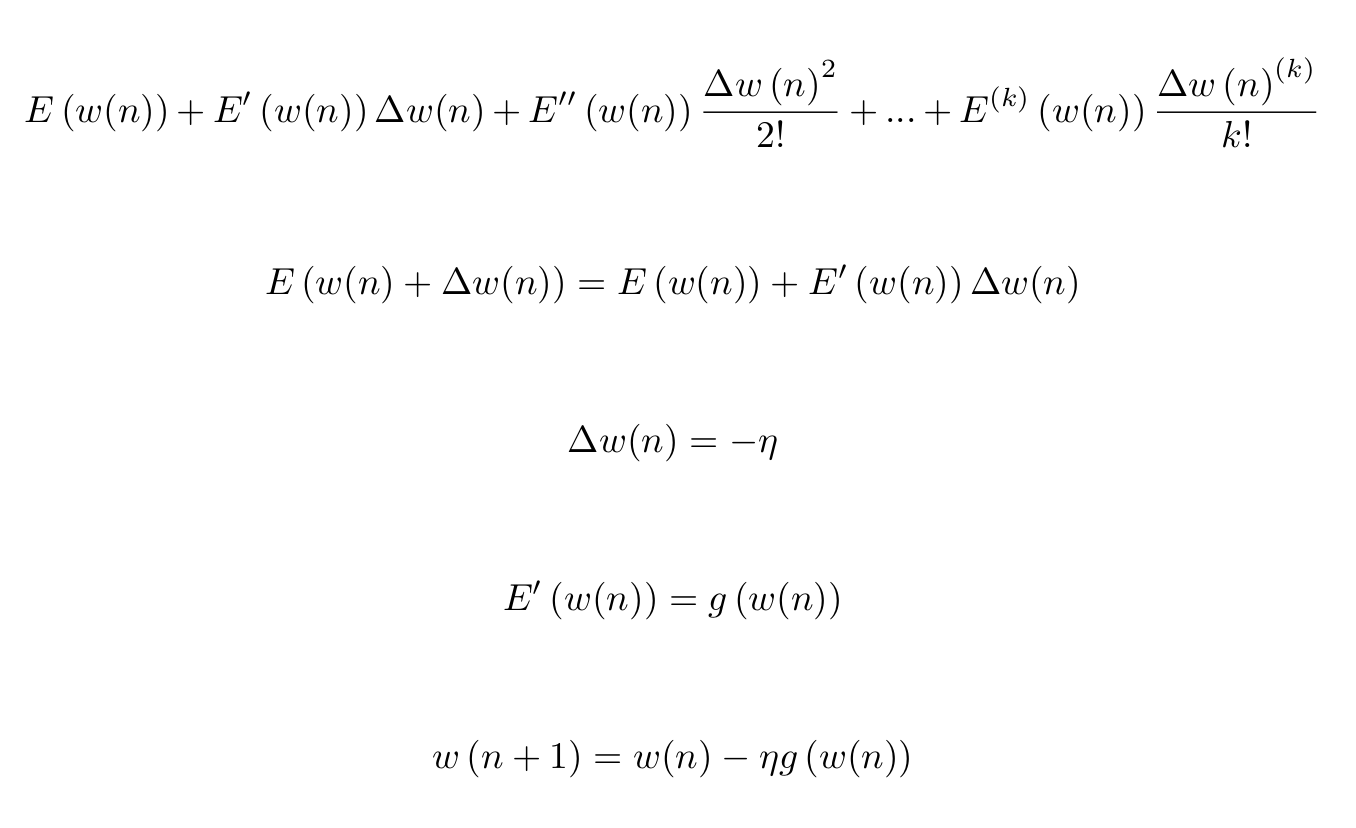
****

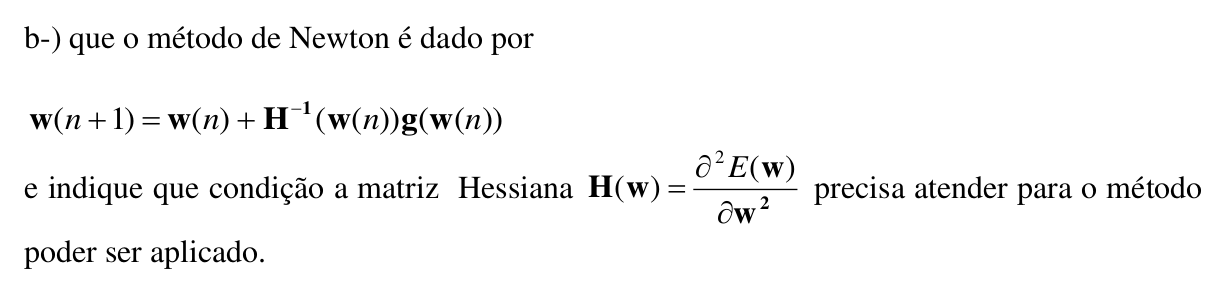
****

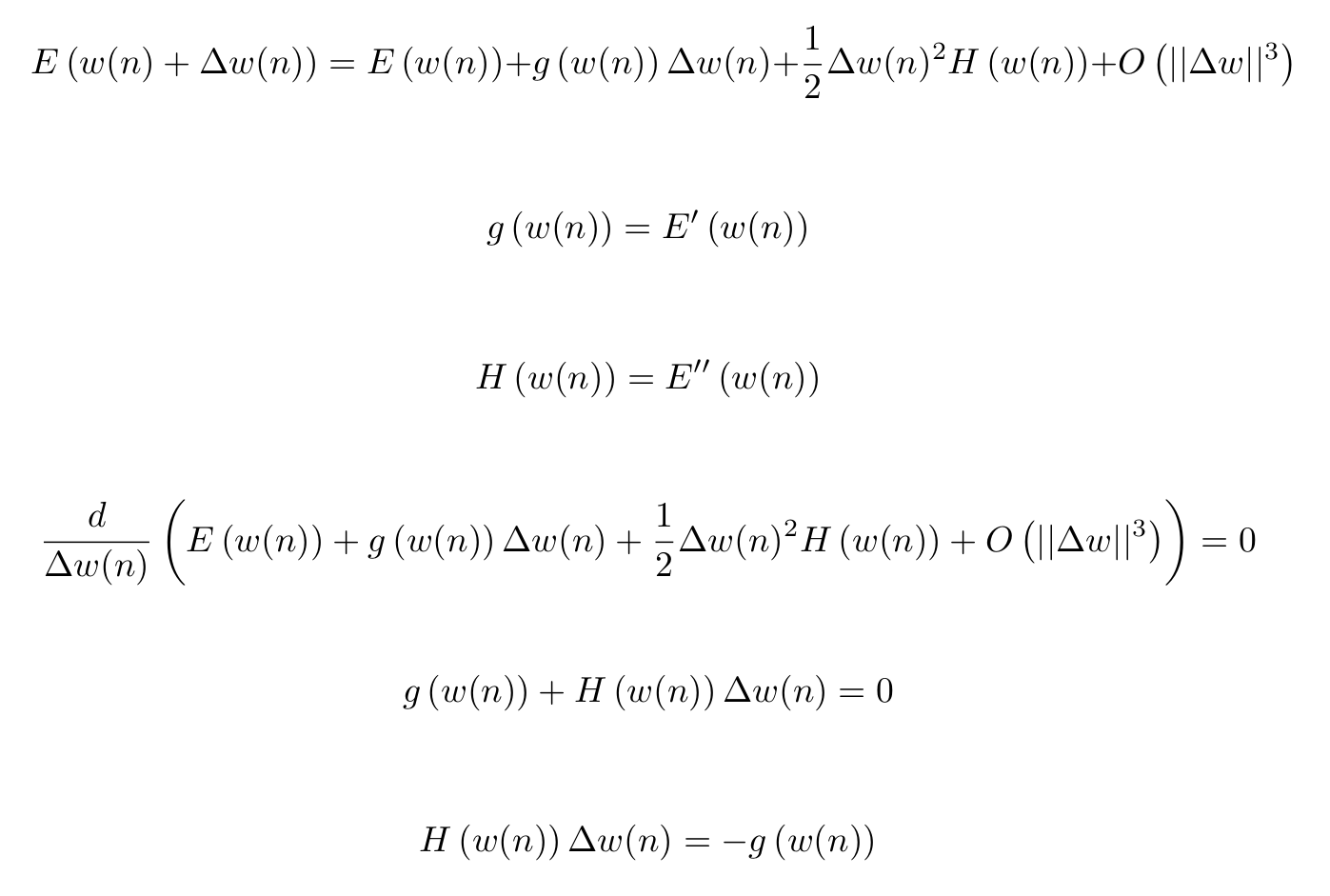
**2º )**

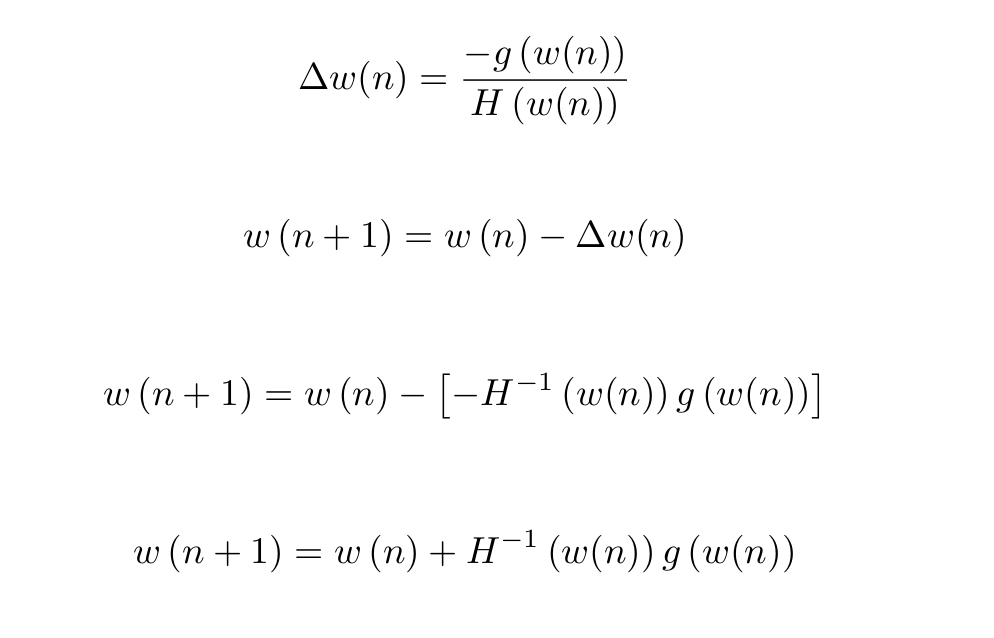
****

****

****

****

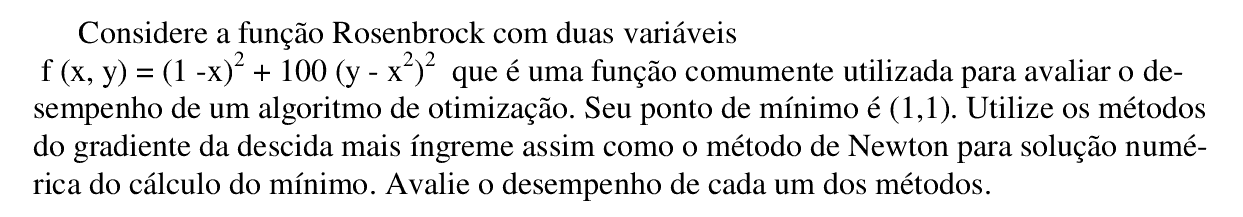
****

****

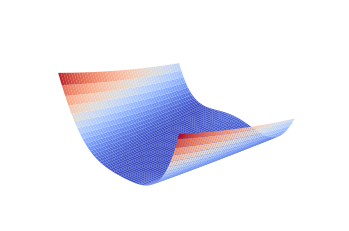
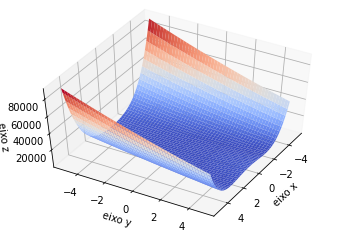
* **Condição da matriz Hessiana para aplicação do método**

O Método de Newton só é apropriado quando o ponto crítico próximo é um mínimo, ou seja, quando os autovalores da matriz Hessiana são positivos.

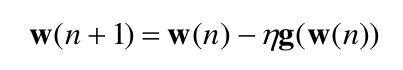
**3º )**

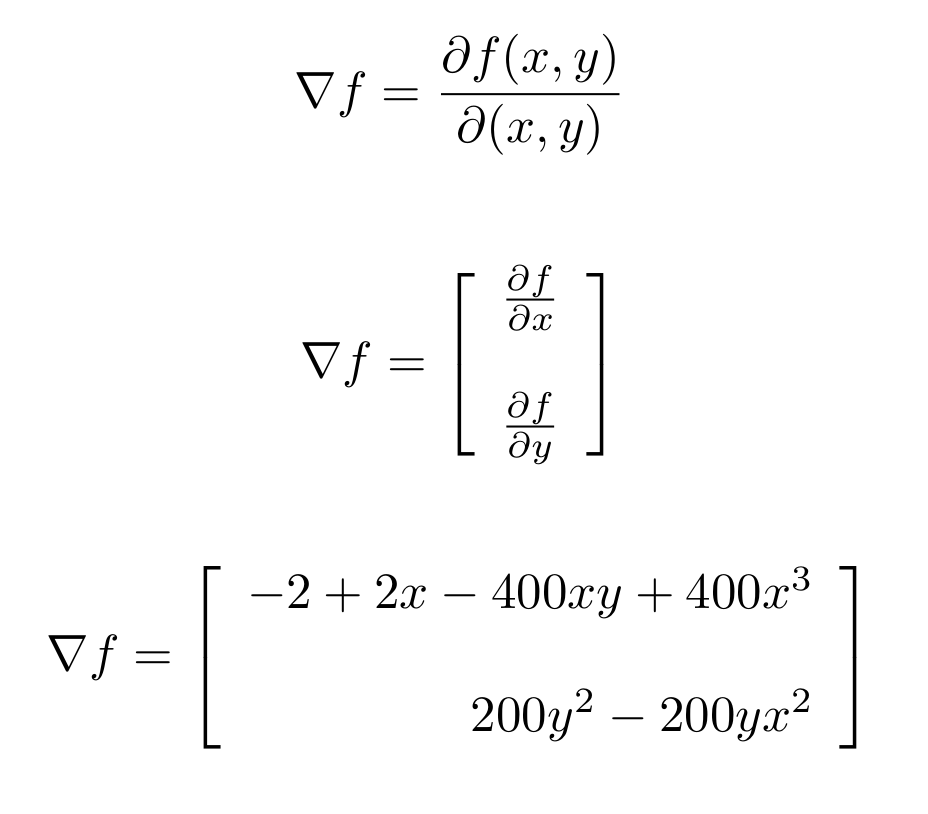
****

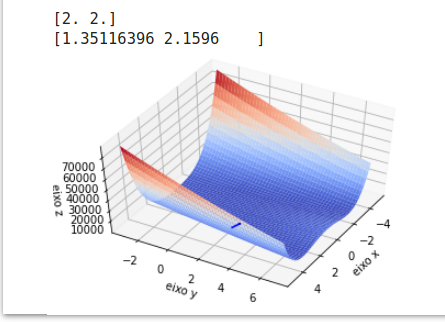
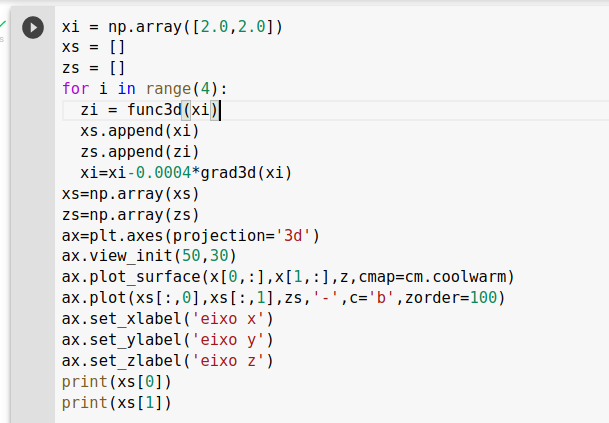
O método de Newton convergiu melhor para o mínimo da função do que o método do gradiente da descida mais íngreme. A função Rosenbrock tem um vale estreito e curvo que contém o mínimo (1,1). Por causa do vale, a otimização está ziguezagueando lentamente com pequenos tamanhos de passo em direção ao mínimo. O método de Newton converge, para o mínimo, mais rapidamente, mas o gradiente descendente escapa melhor de pontos de sela.



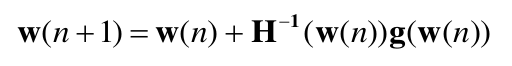
* **Método do gradiente da descida mais íngreme**

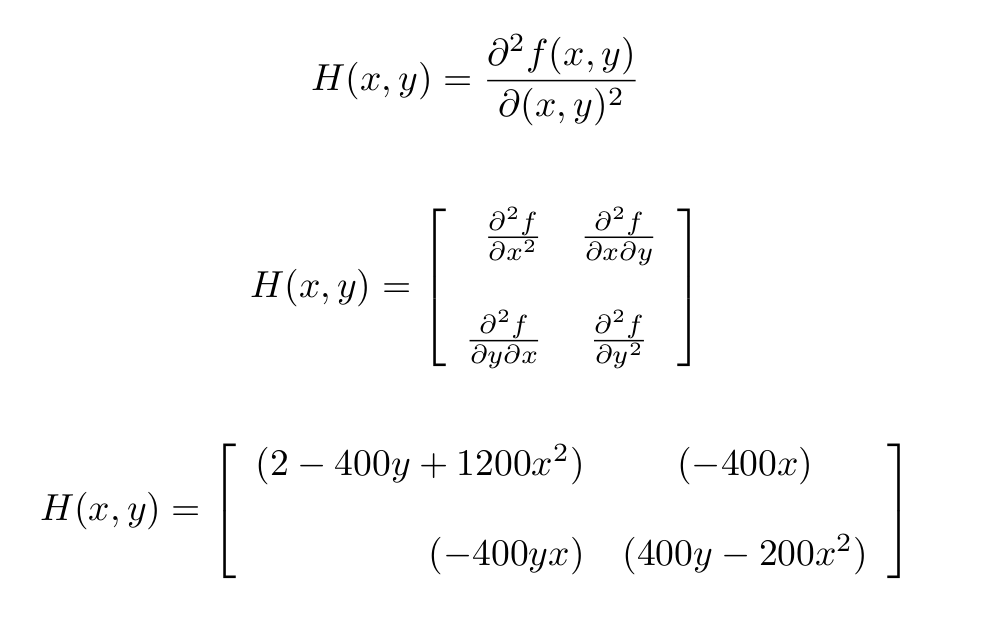
****

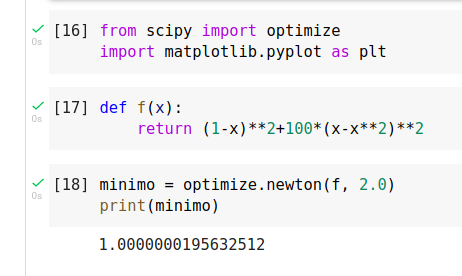
****

****

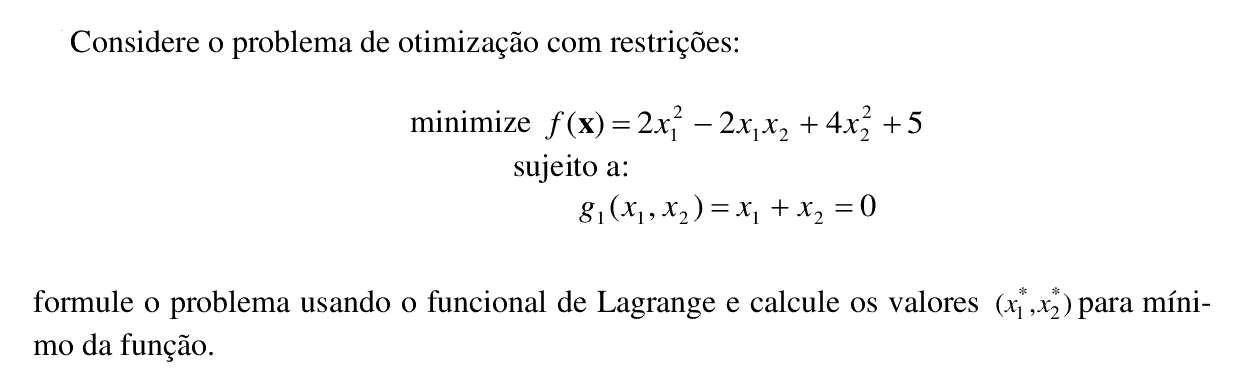
* **Método de Newton**

****

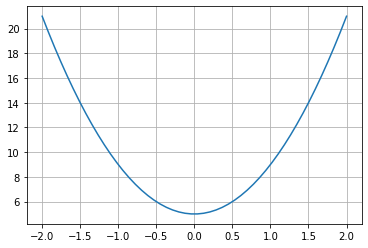
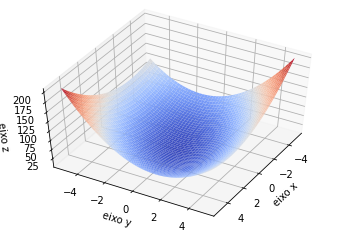
****

****

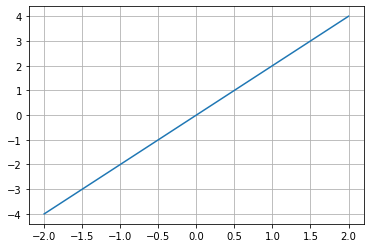
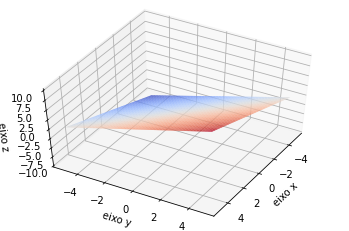
**4º )**

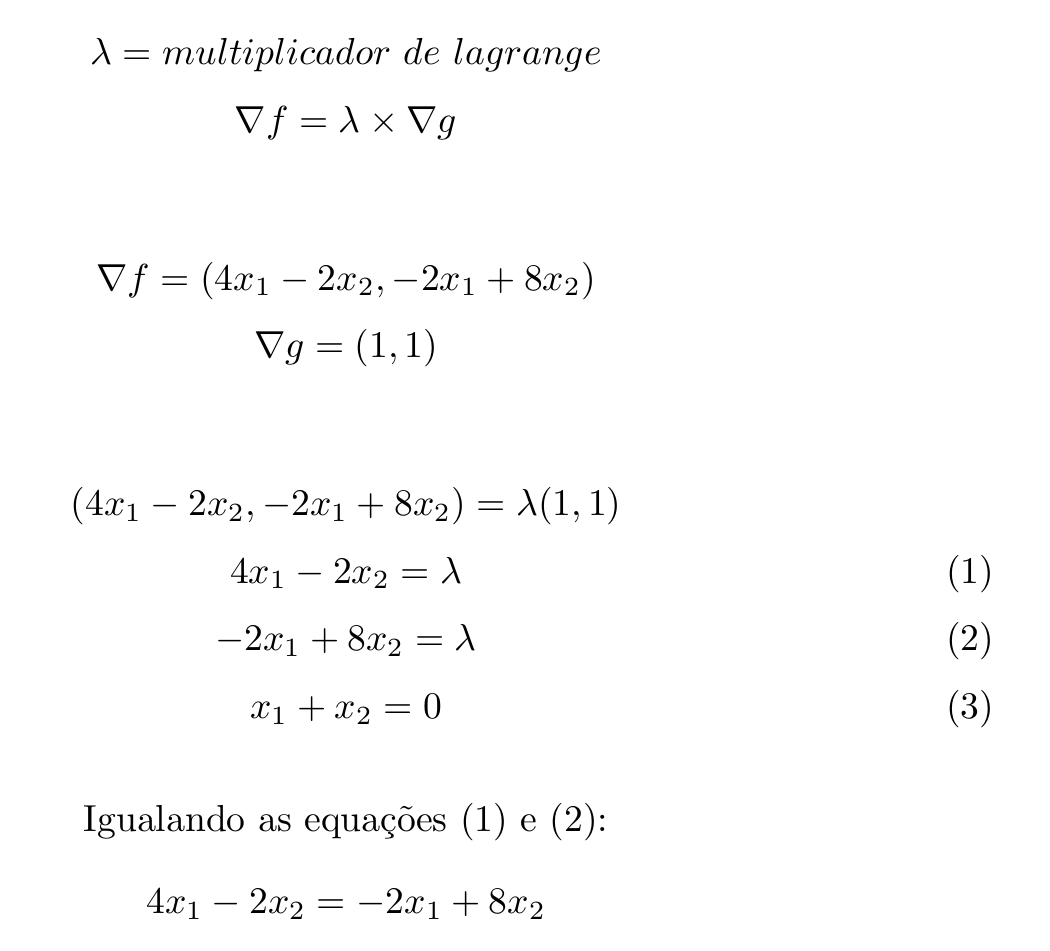
****

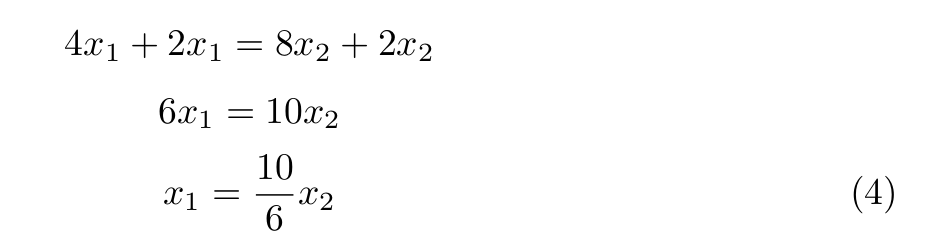
* **Função f(x):**

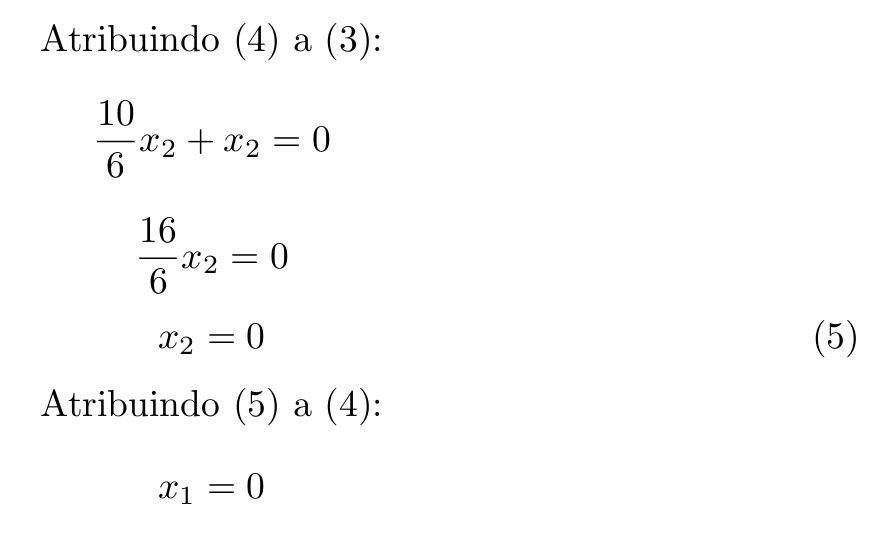
****

* **Função de restrição g(x):**

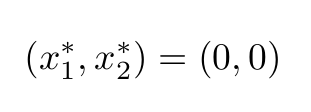
****

****

****

****

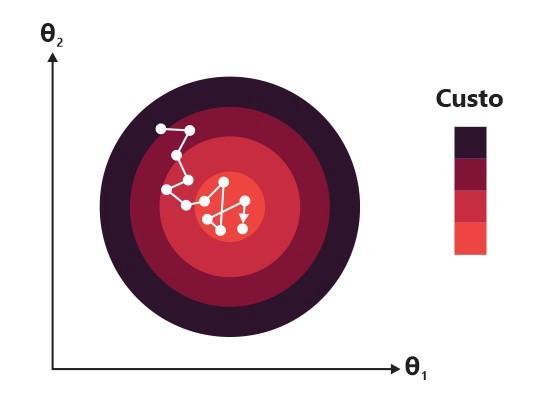
**Mínimo da função:**

****

**5º )**

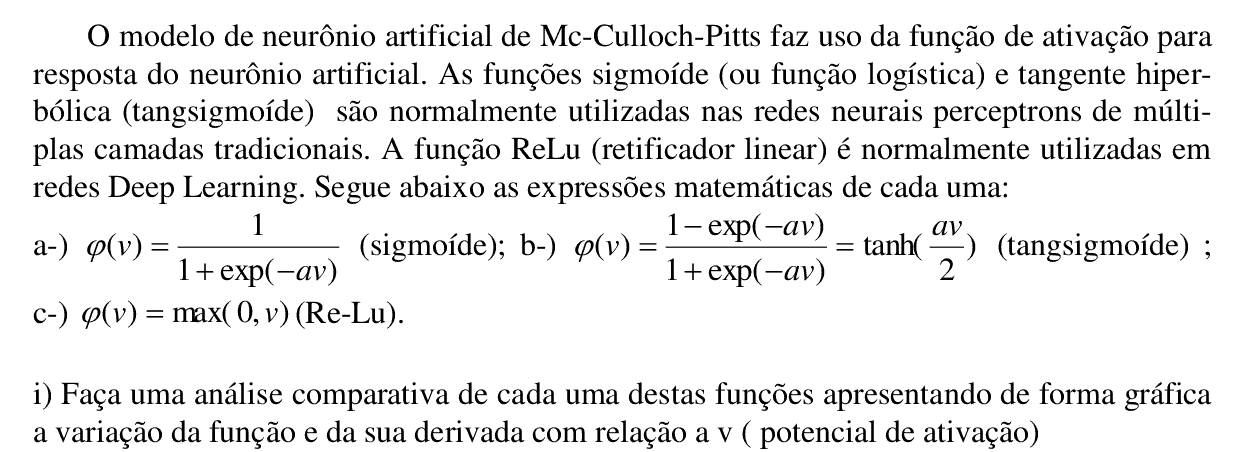
****

O gradiente estocástico (SGM) é um método de otimização utilizado em aprendizado de máquina. Estocástico, nesse contexto, pode ser entendido como aleatório. O SGM, de forma aleatória, escolhe uma instância do conjunto de treinamento com a finalidade de realizar o cálculo do gradiente se baseando nessa instância. Desta forma, o algoritmo se torna rápido devido a pequena quantidade de dados para manipular em cada iteração. O SGM permite realizar o treinamento com grandes volumes de dados. A função custo não diminui suavemente até o mínimo, ela oscila e diminui na média, ou seja, os valores obtidos não são ótimos.



A aleatoriedade, desse método, funciona bem para escapar de ótimos locais, mas existe a possibilidade do algoritmo nunca se estabelecer no objetivo, que é o mínimo global. Reduzir gradualmente a taxa de aprendizado, funciona como uma estratégia para fugir desse problema. As etapas começam com uma taxa de aprendizagem alta para fugir dos mínimos locais e diminuem para atingir o mínimo global. Porém, se a taxa for reduzida rapidamente, o algoritmo poderá ficar preso em um mínimo local. Se reduzida lentamente, poderá saltar em torno do mínimo e obter uma solução insuficiente. Cronograma de aprendizado (learning schedule) é a função responsável por definir a taxa de aprendizado para cada iteração.

**6º )**

****

* **Funções de ativação:**

São elementos, das redes neurais, responsáveis decidir se um neurônio deve ser ativado ou não. Ou seja, se a entrada que o neurônio está recebendo é relevante para a informação fornecida ou deve ser ignorada. Limita a saída de um neurônio em um intervalo de valores.

* **Combinador Linear:**

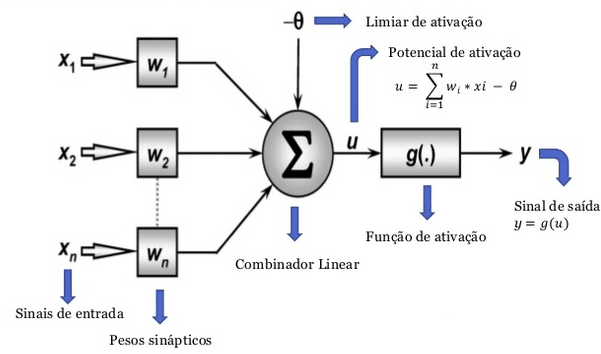
Agregar os sinais de entrada ponderados pelos pesos sinápticos com a finalidade de gerar um potencial de ativação.

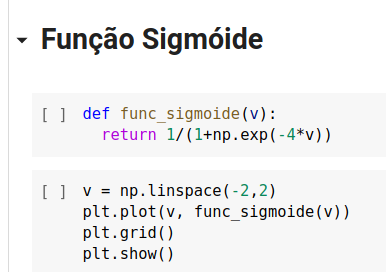
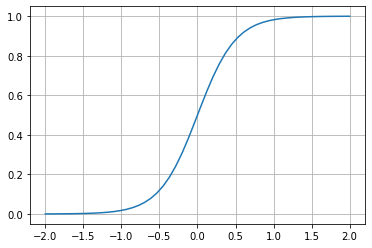
* **Limiar de ativação:**

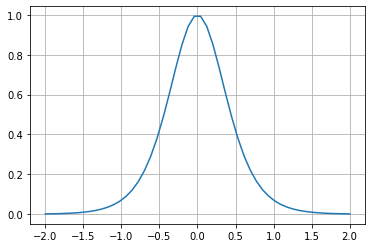
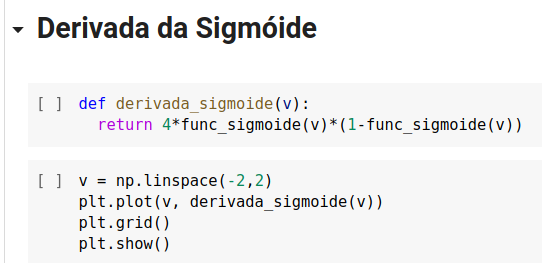
Faz com que o resultado produzido pelo combinador linear possa gerar um valor de disparo de ativação adequado.

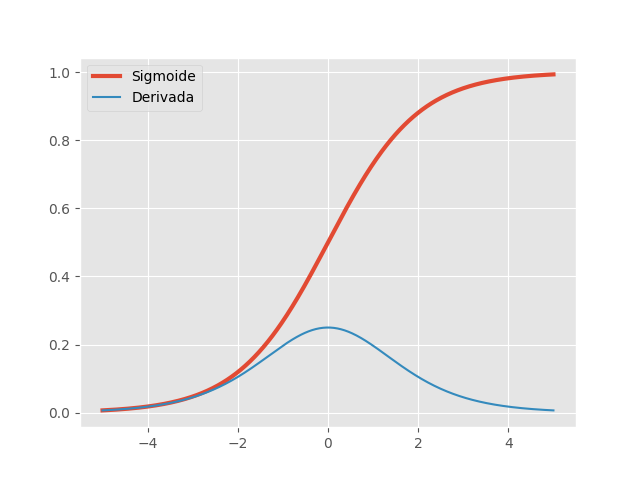
* **Potencial de ativação (v):**

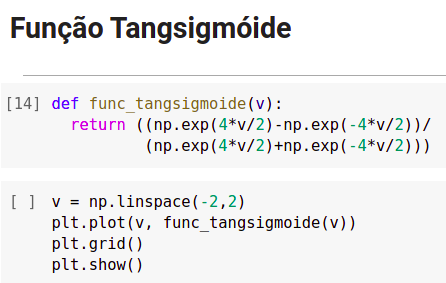
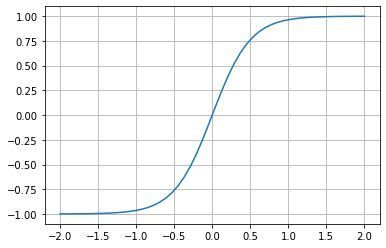
É o resultado obtido pela diferença do valor produzido entre o combinador linear e o limiar de ativação. Se o valor for positivo, ou seja, se v ≥ 0 então o neurônio produz um potencial excitatório; no caso contrário, o potencial se configura como inibitório.

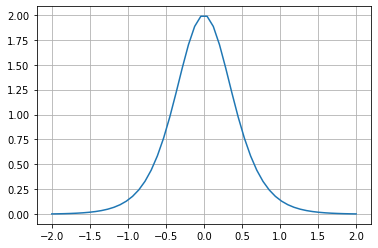
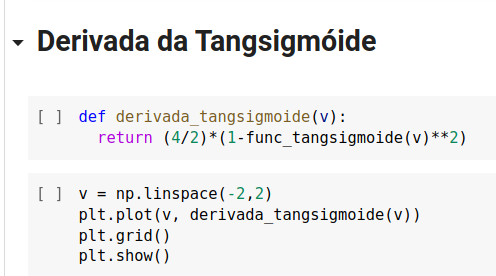


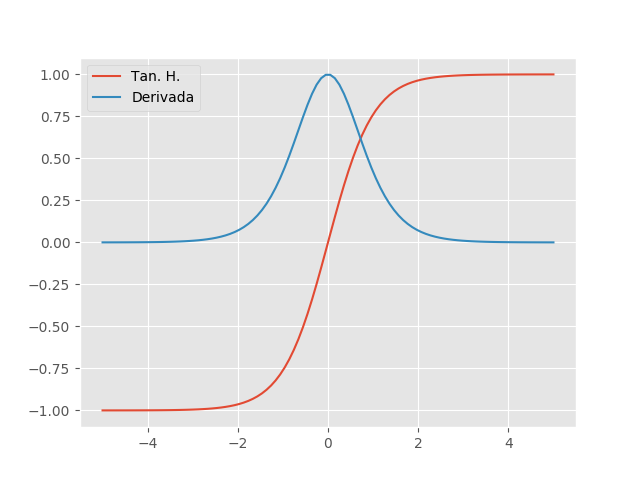
** **

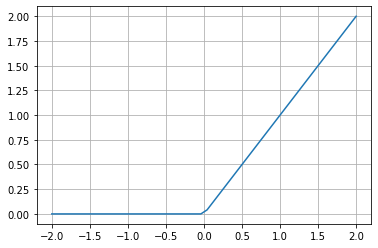
****

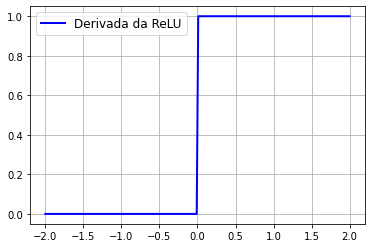
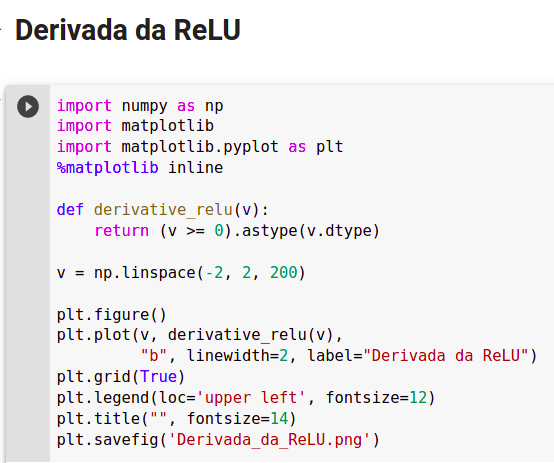
****

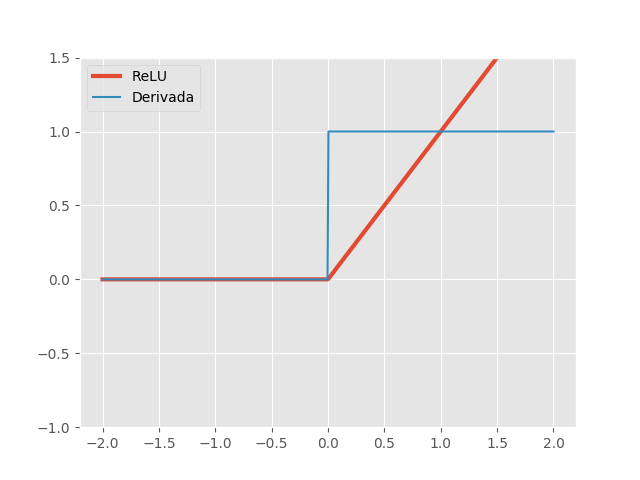
** **

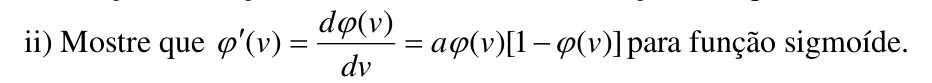
****

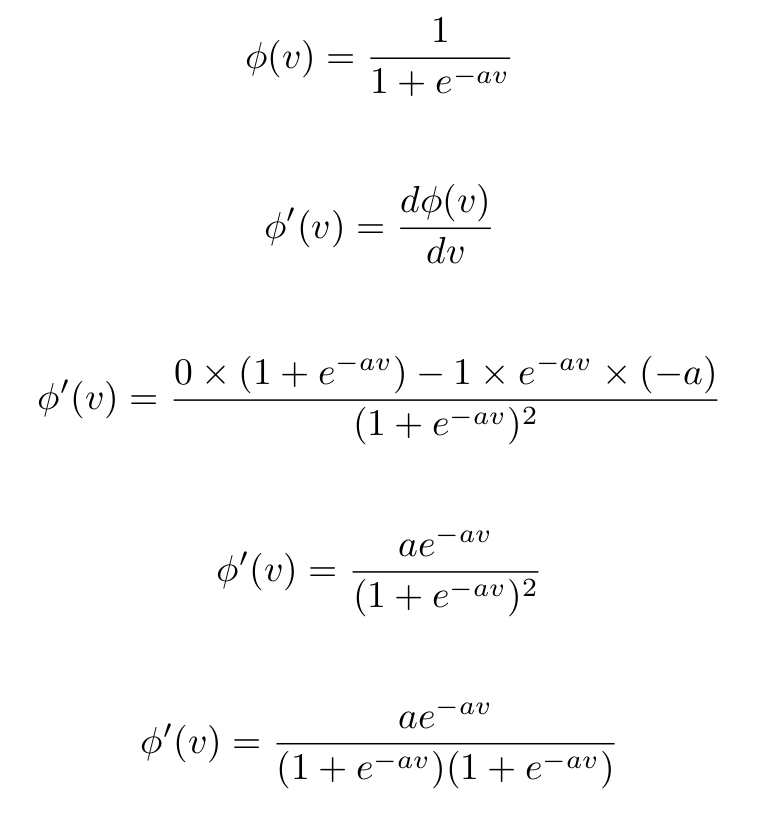
****

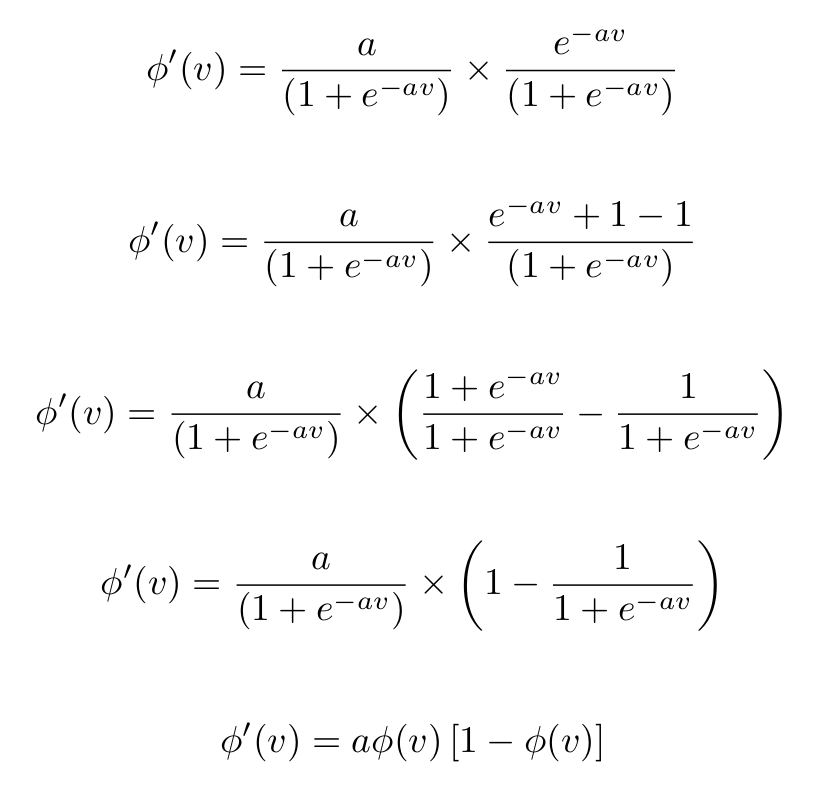
****

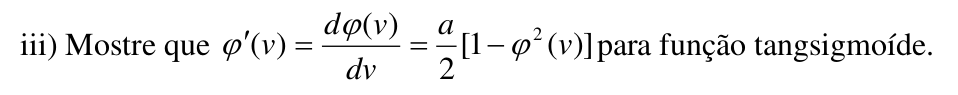
****

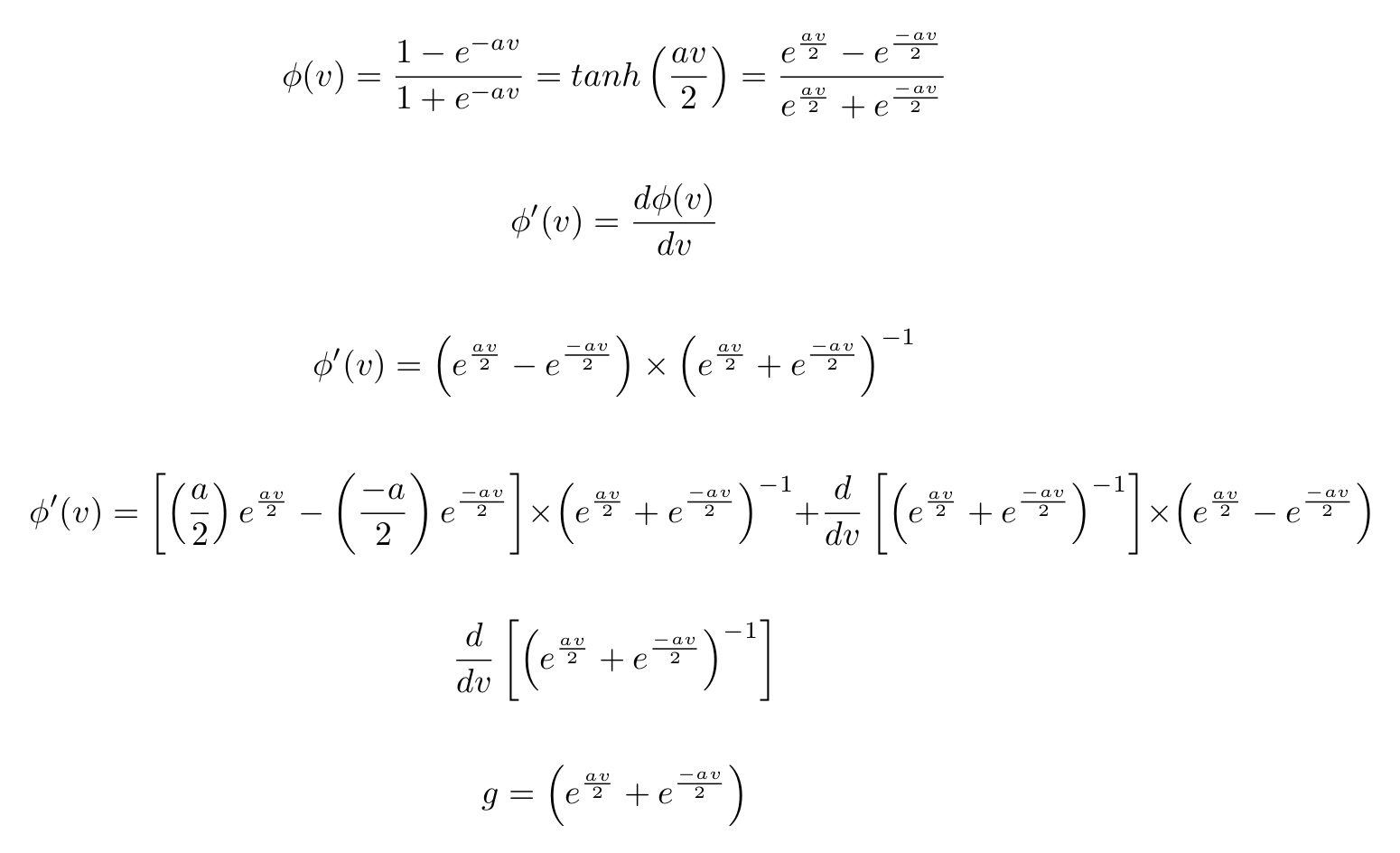
****

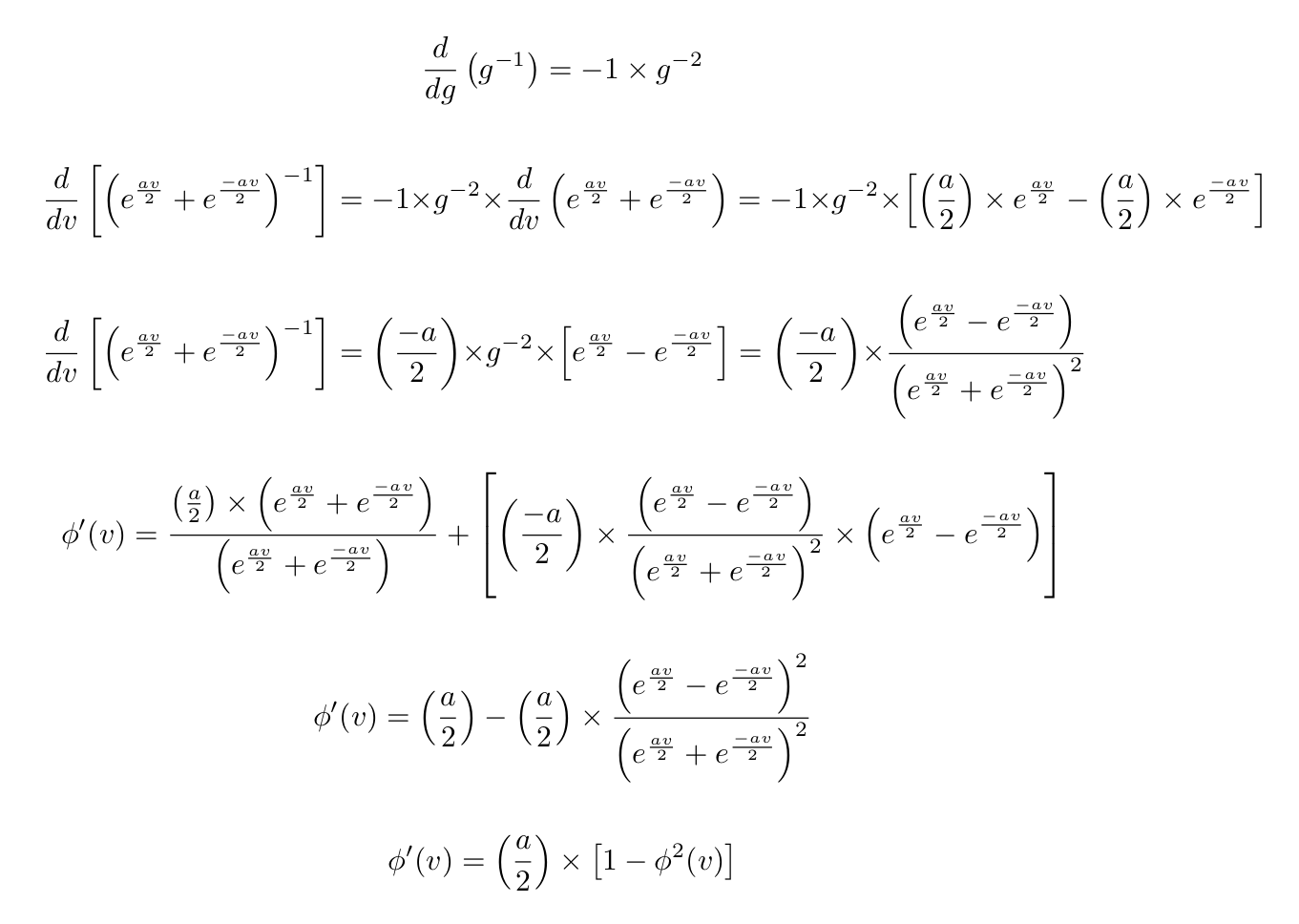
****

****

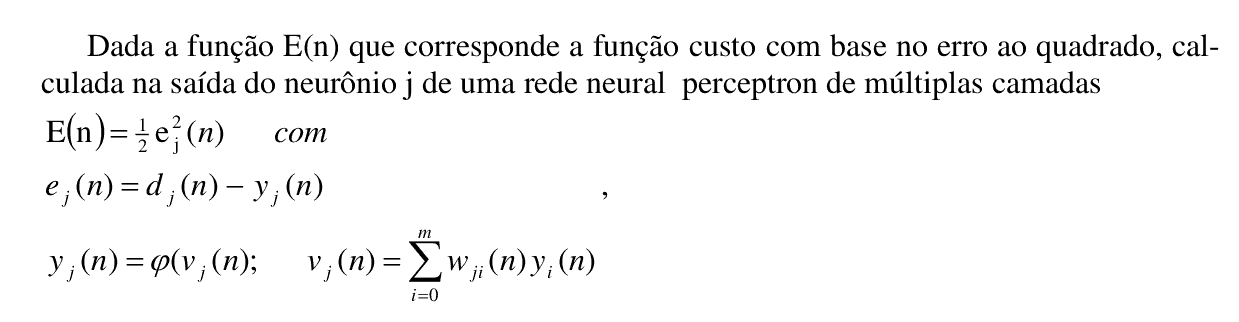
****

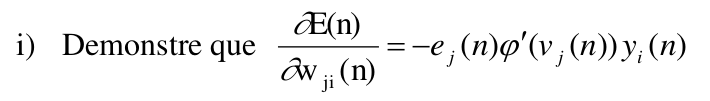
****

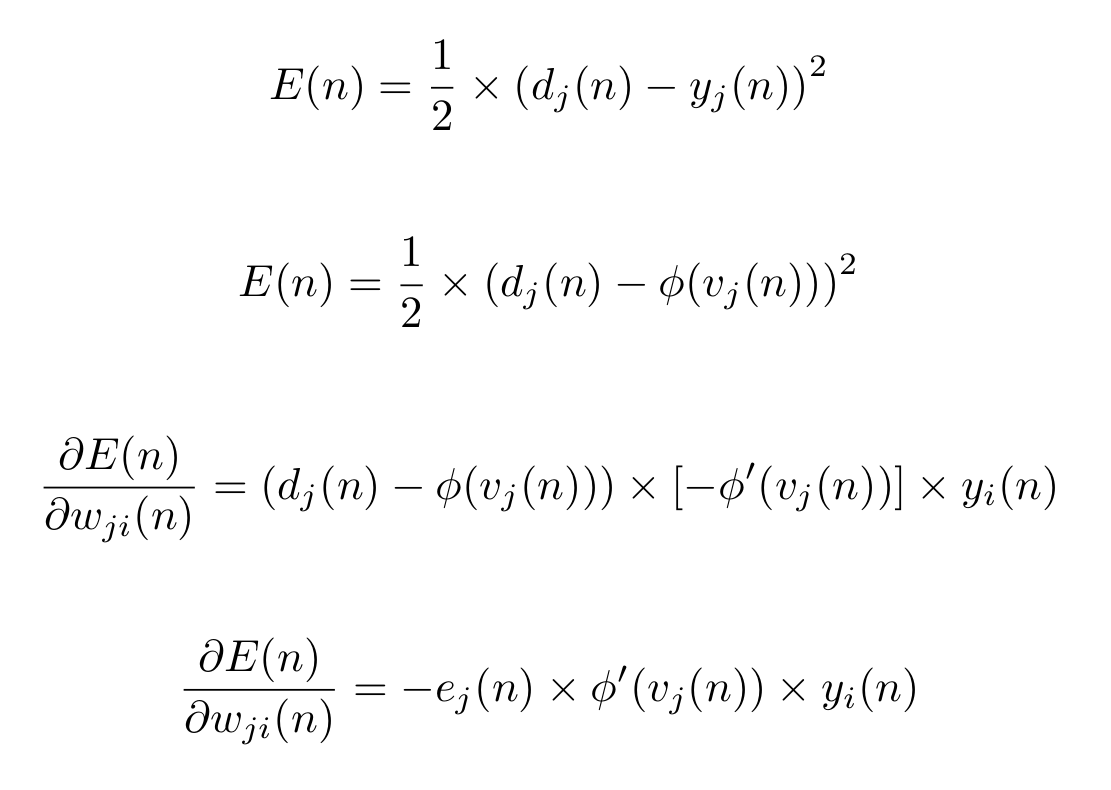
****

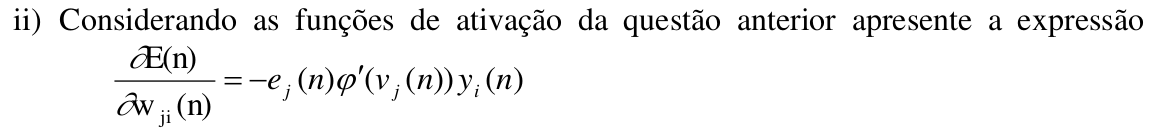
****

**7º )**

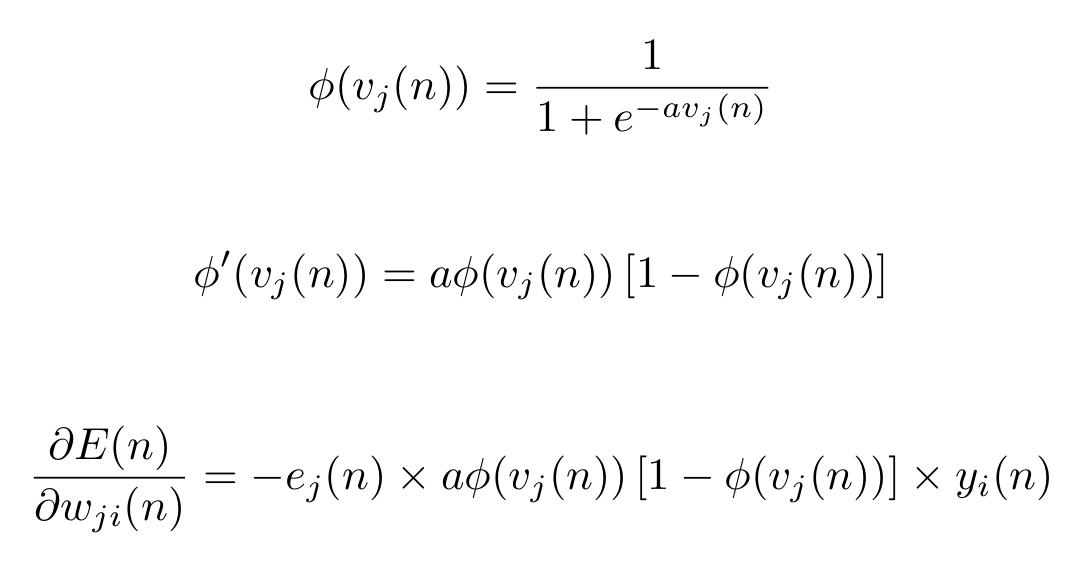
****

****

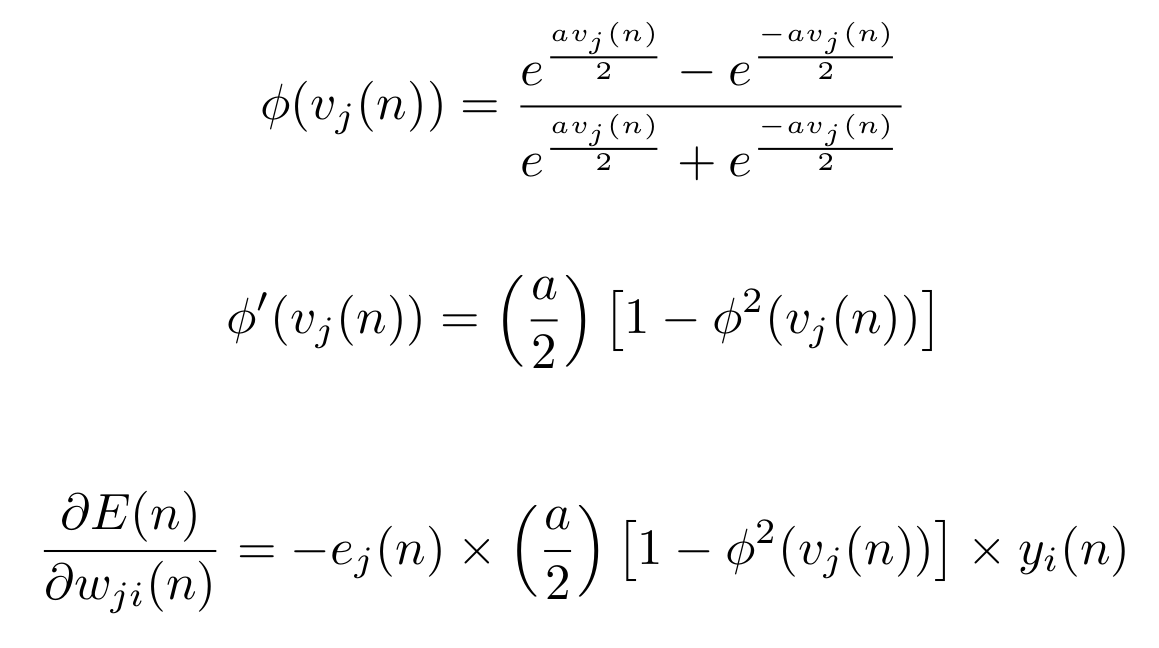
****

****

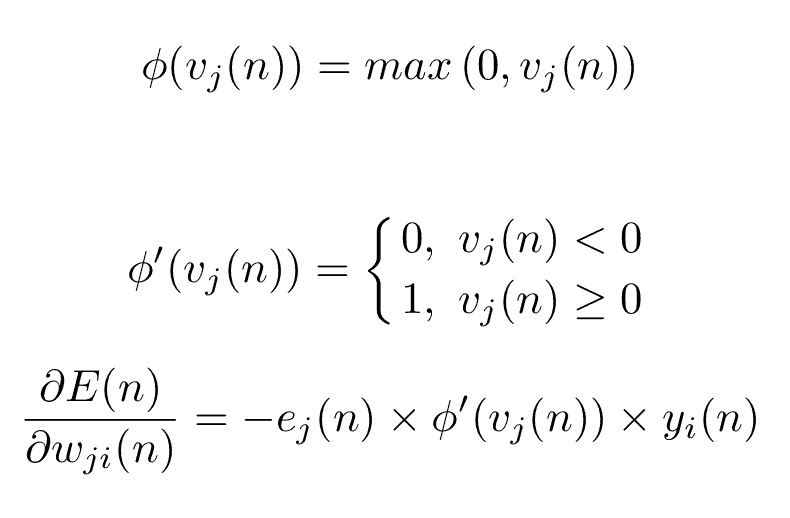
* **Para a função sigmóide:**

****

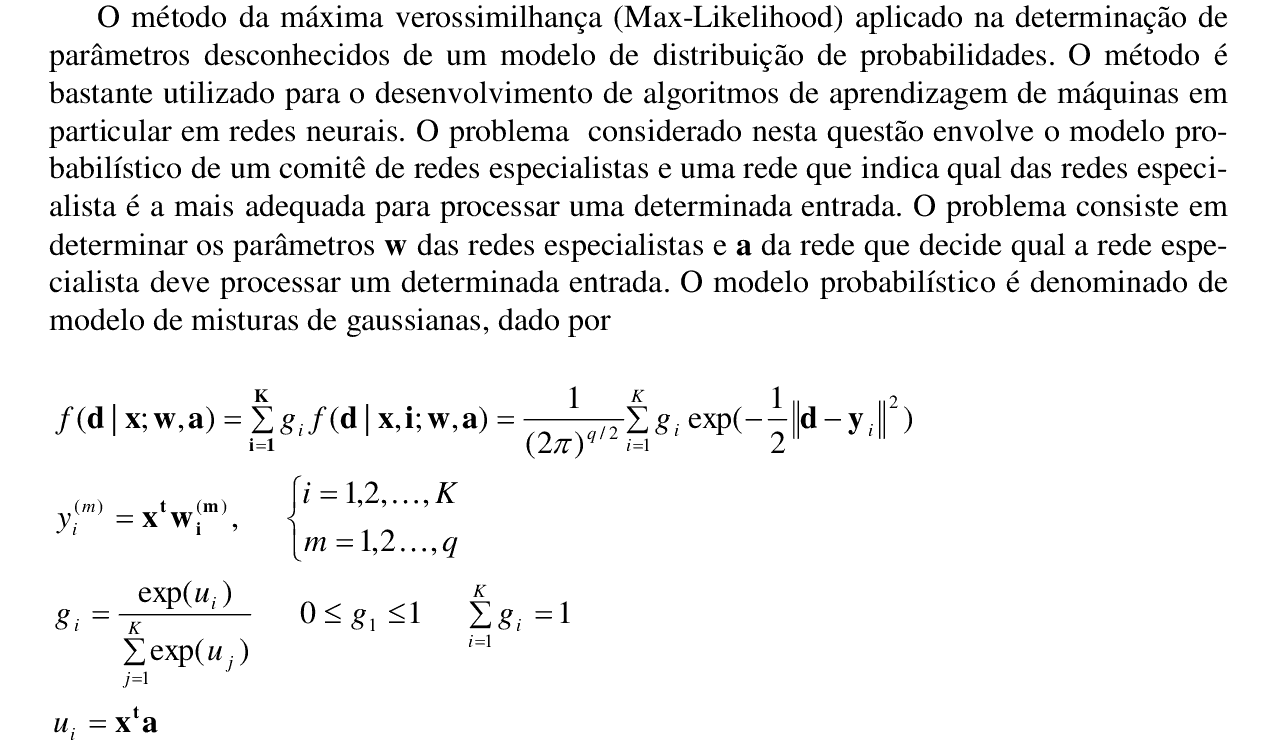
* **Para a função tangsigmóide:**

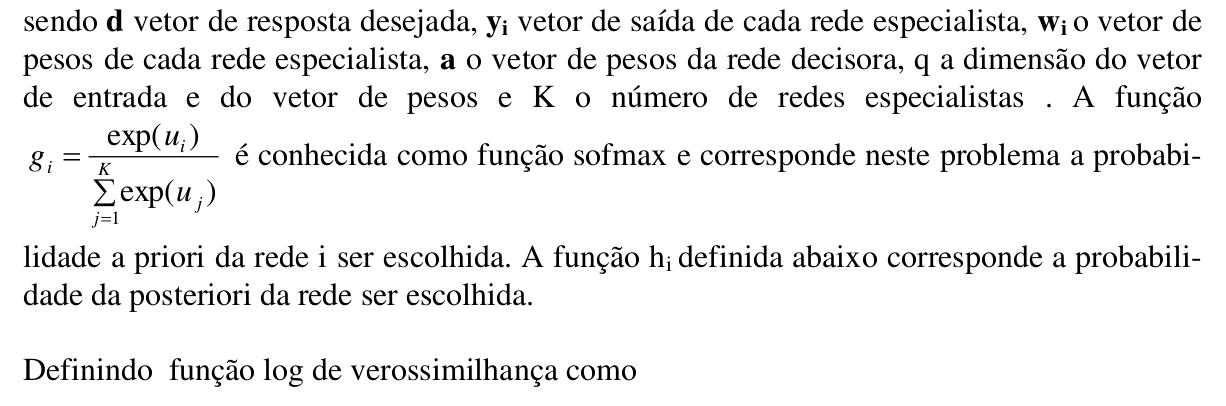
****

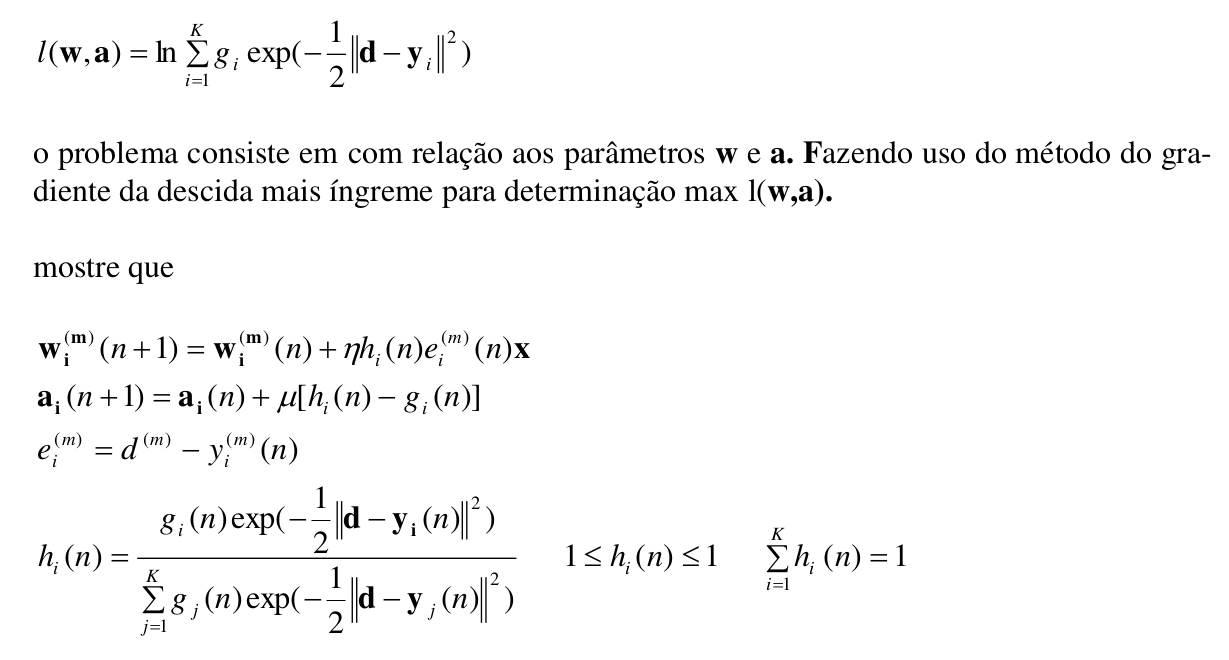
* **Para a função Re-Lu:**

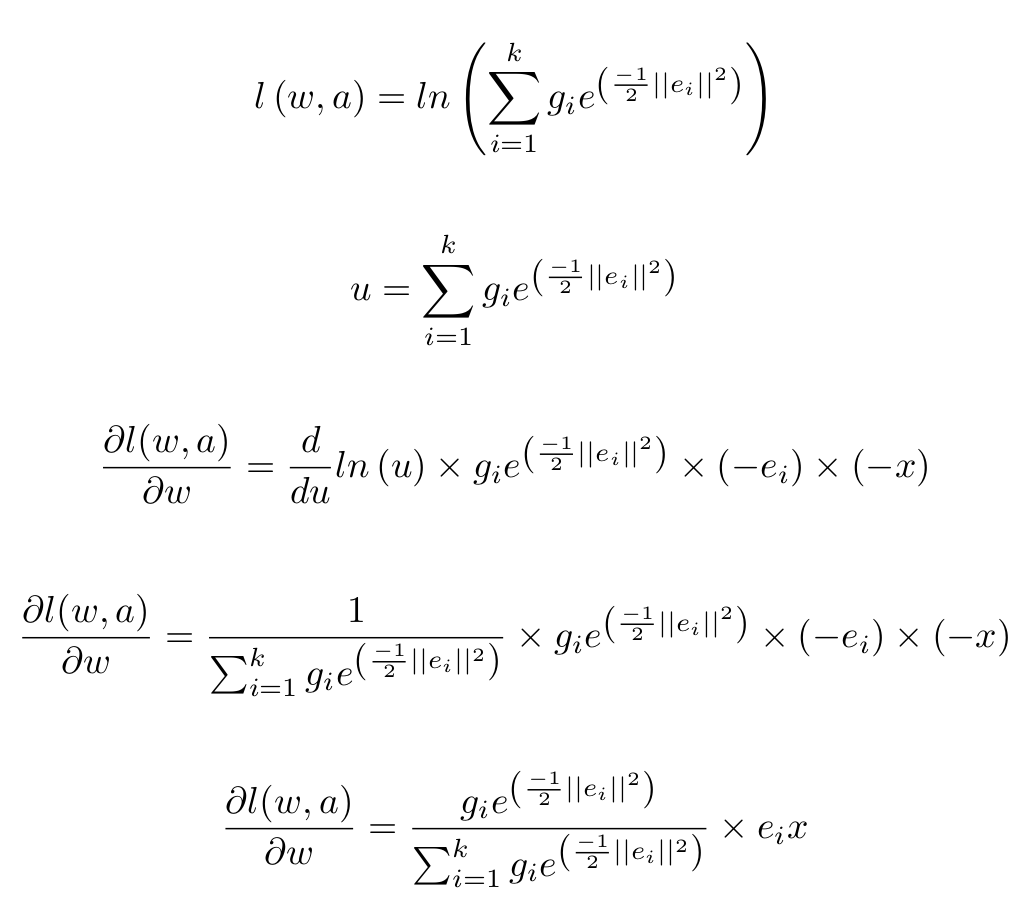
****

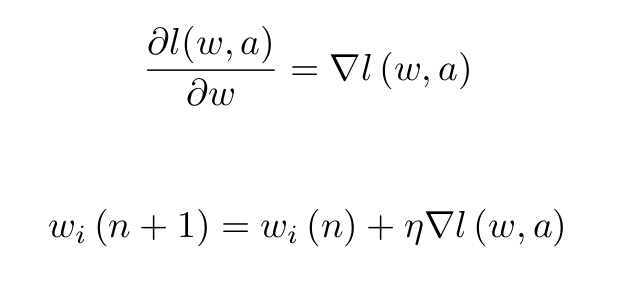
**8º )**

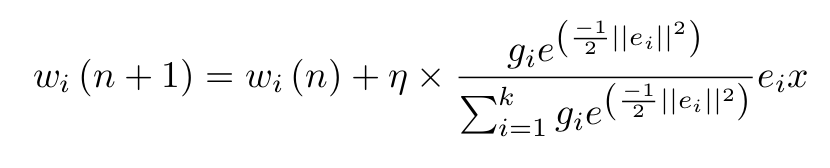
****

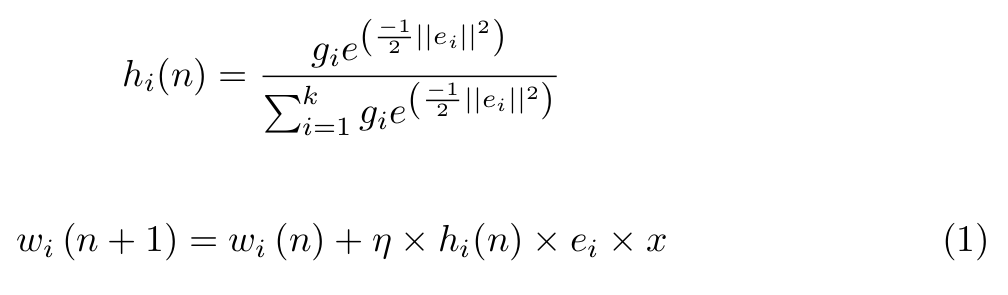
****

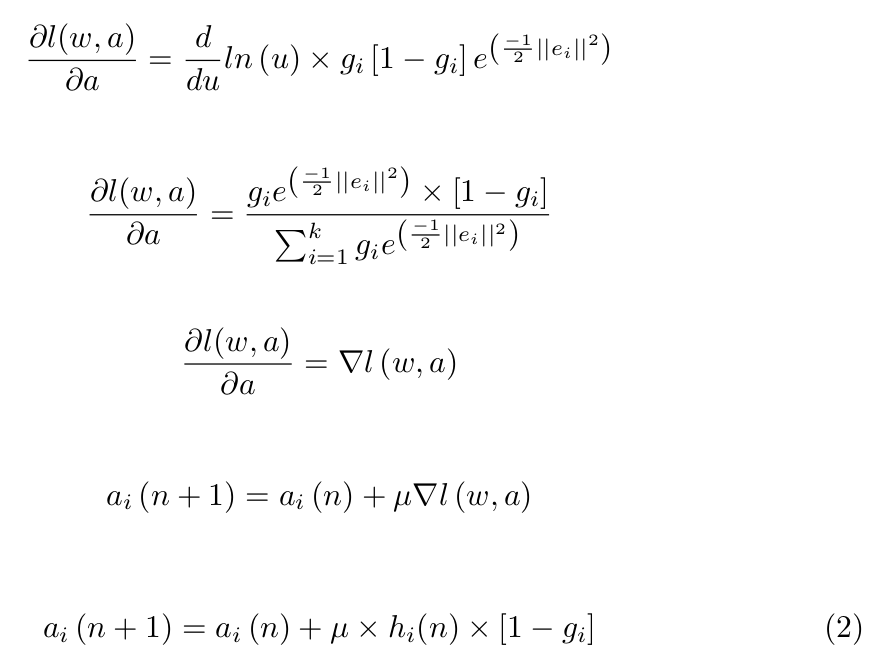
****

****

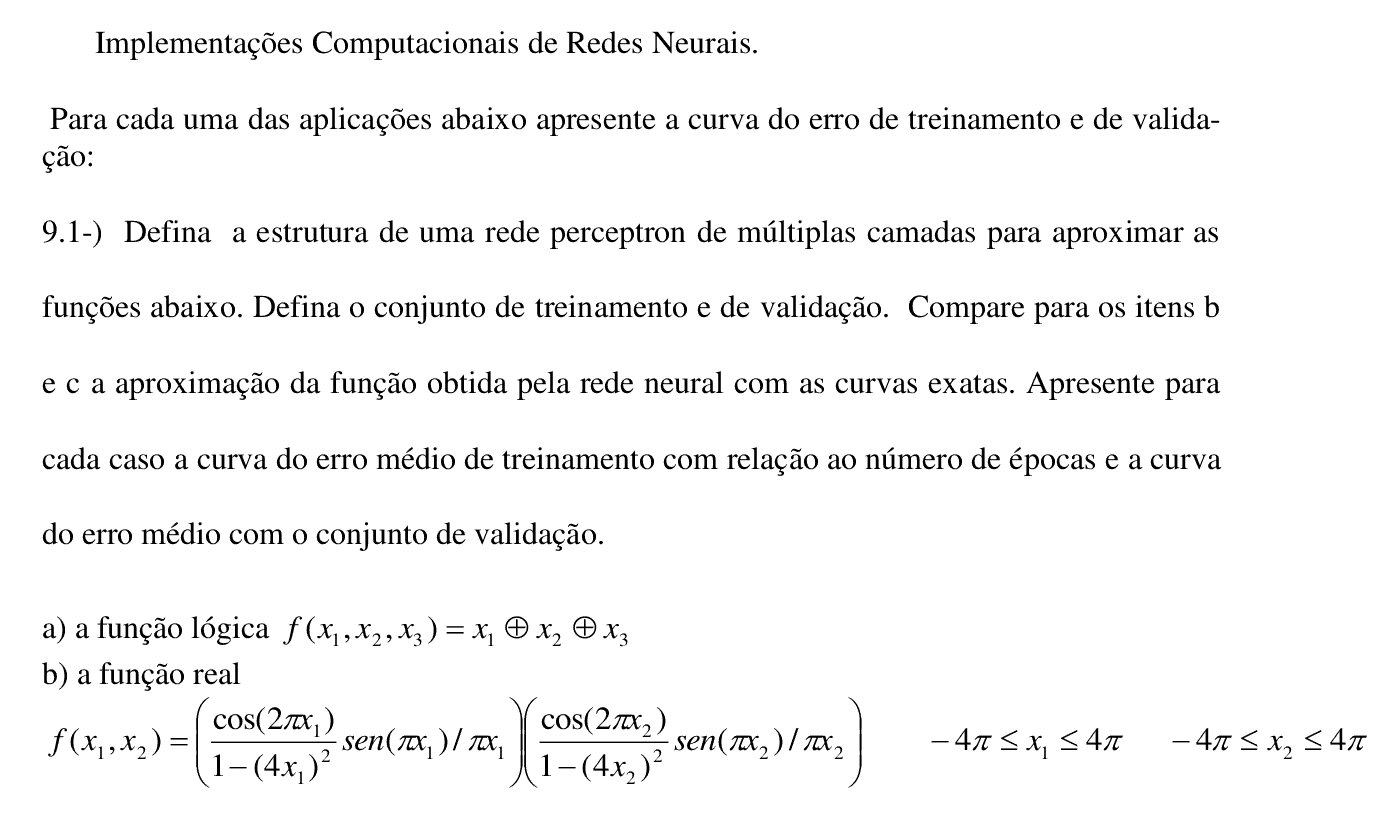
****

****

****

****

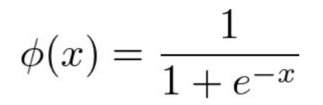
**9º )**

****

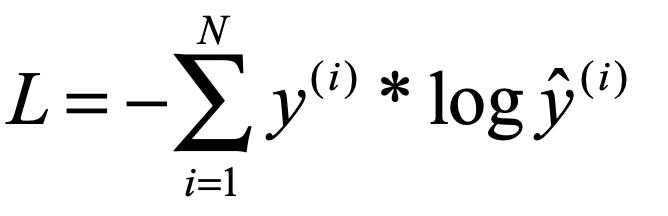
**Épocas:** número de vezes que um algoritmo de machine learning analisa um conjunto de dados completo. No caso do uso de minilotes o número de épocas não é igual ao número de interações.

**Função de perda (loss function):** calcula a diferença entre a saída desejada e a saída atual. Informa o nível de precisão da rede neural fazendo previsões para as entradas.

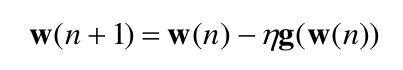
**Activation function:** Sigmóide

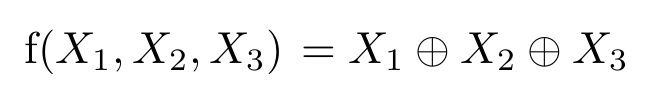


**Loss function:** Entropia cruzada

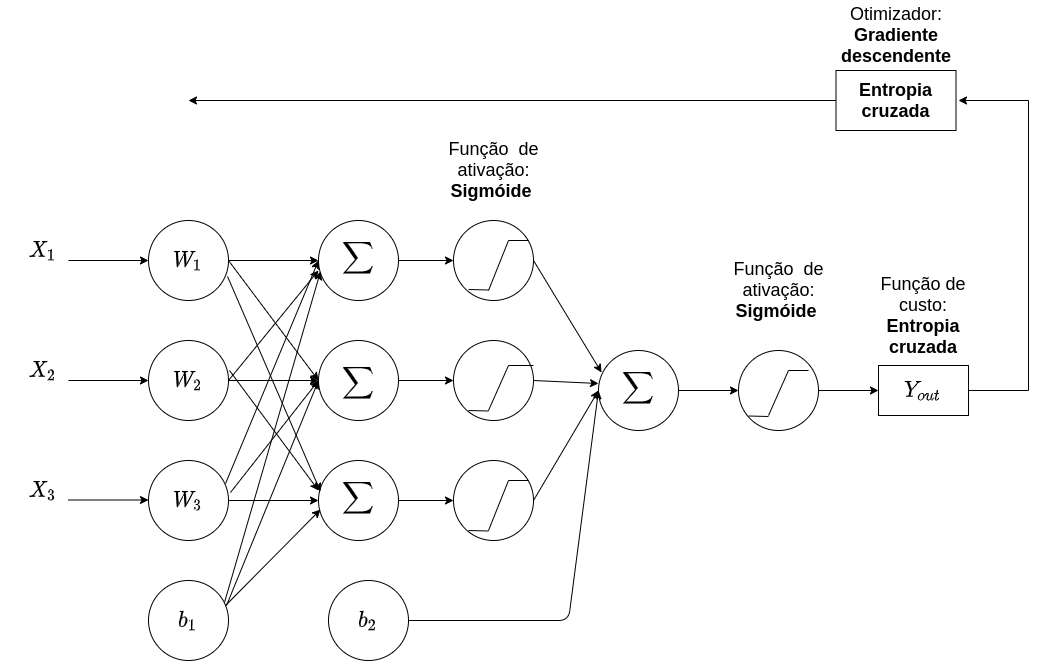


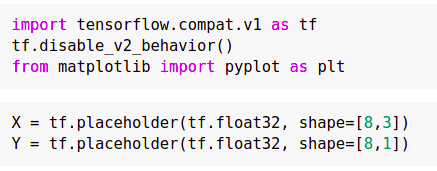
**Optimizer:** Gradient Descent

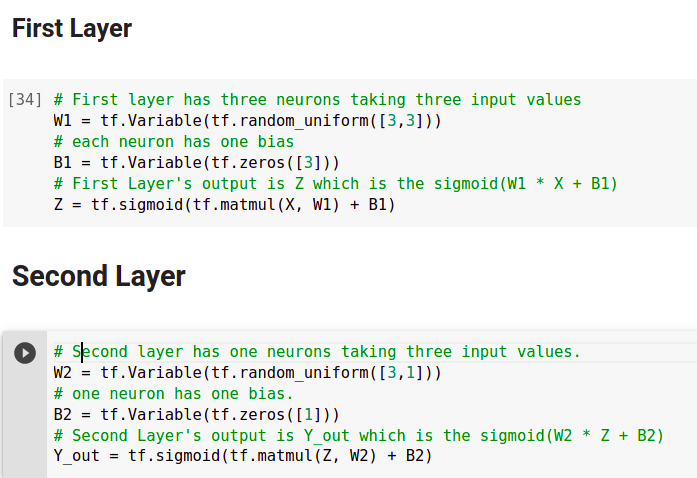
****

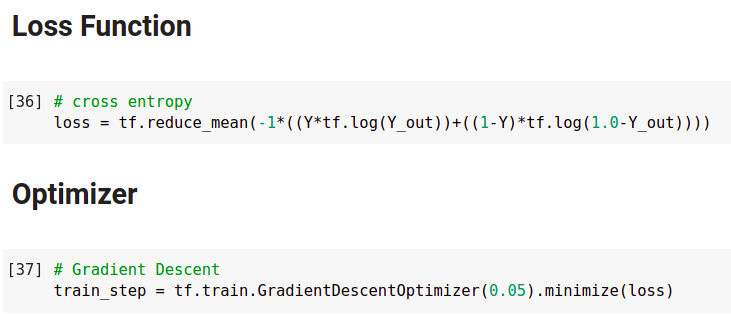
****

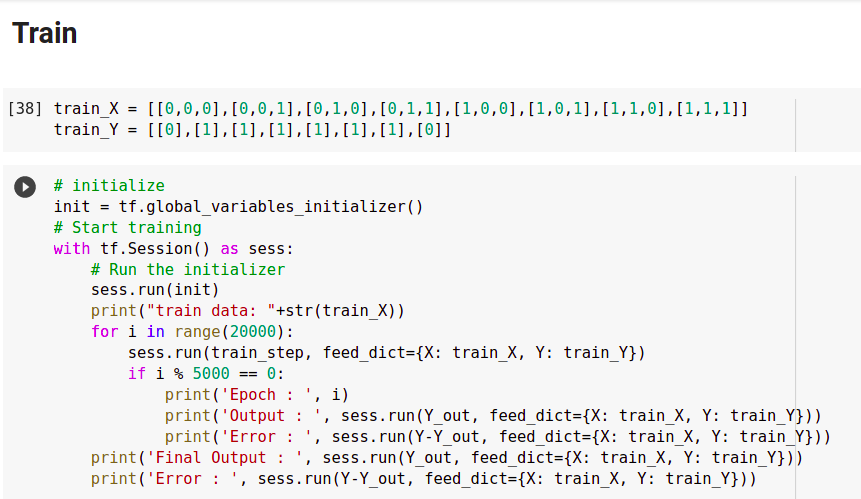
| X1 | X2 | X3 | Y |
| --- | --- | --- | --- |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

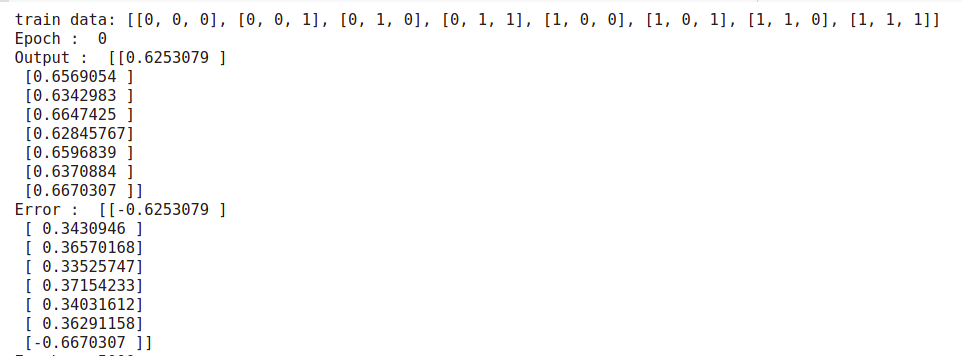
****

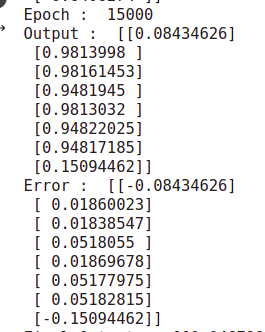
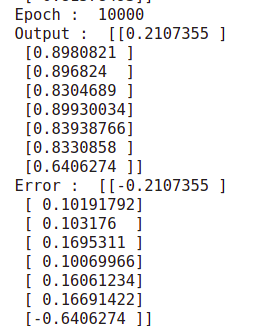
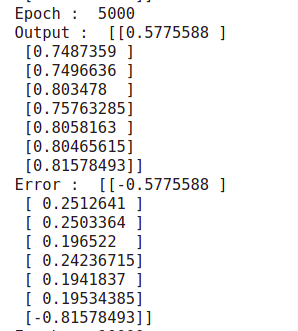
****

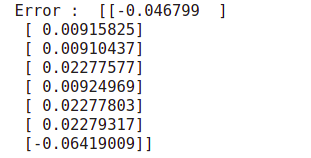
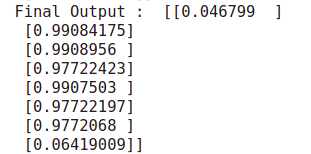
****

****

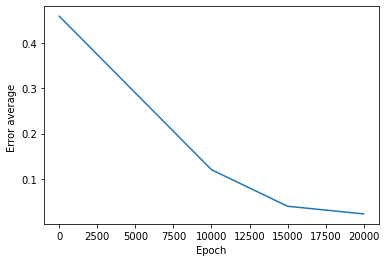
****

****

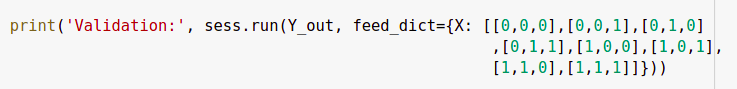
****

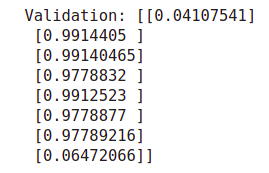
****

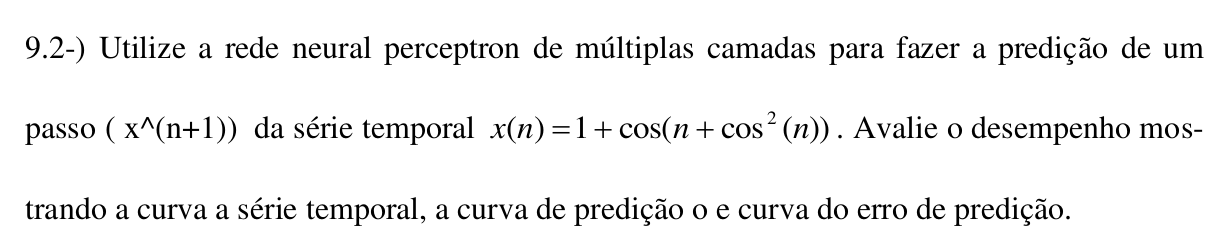
* **Curva do erro médio de treinamento com relação ao número de épocas**

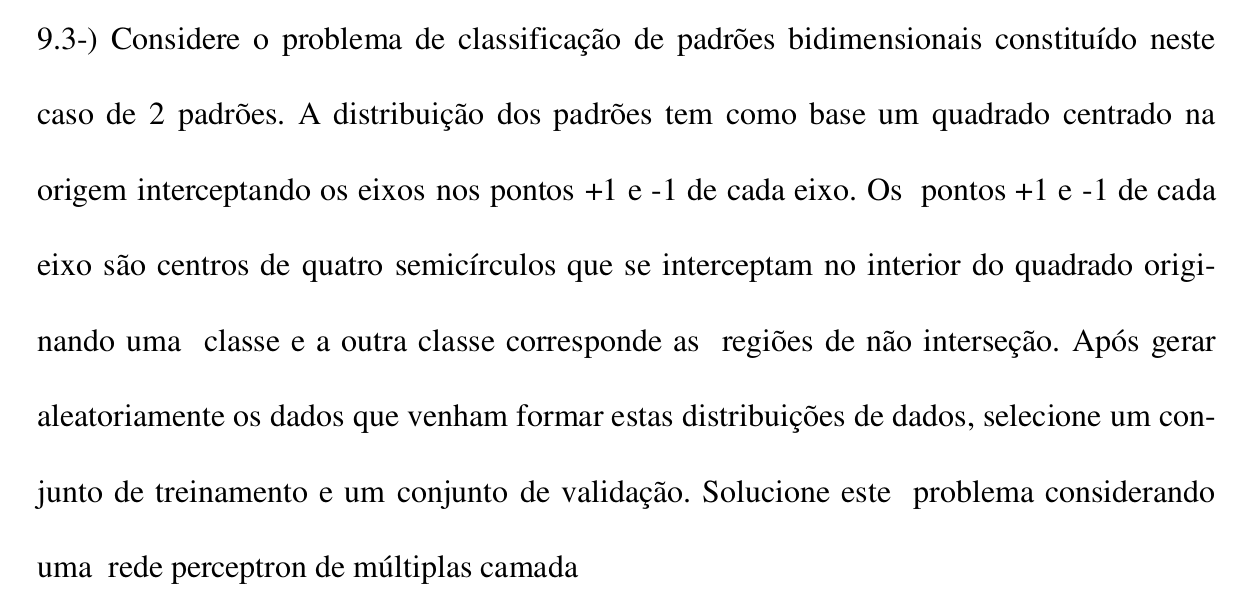
****

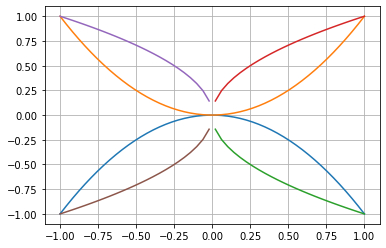
* **Validação (teste):**

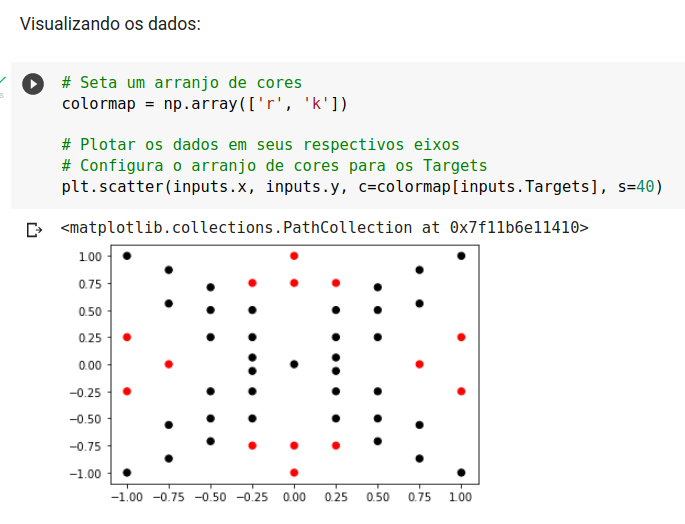
****

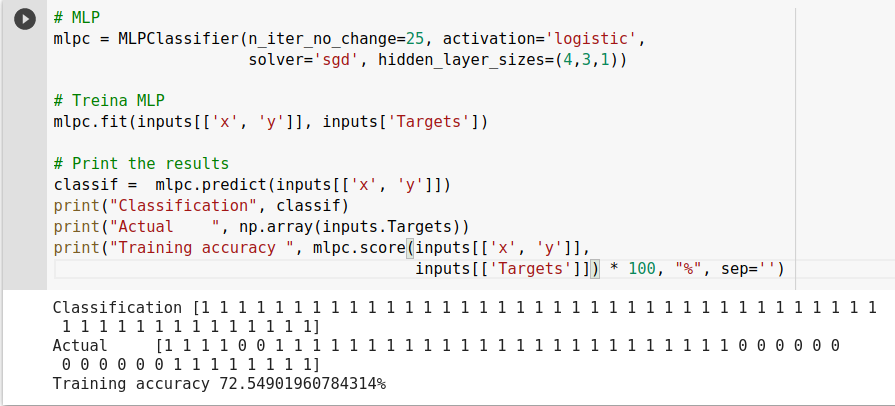
****

****

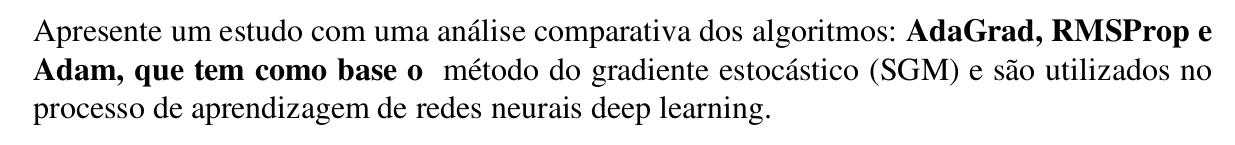
****

****

****

****

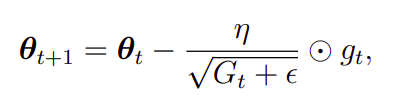
**10º ) Trabalho:**

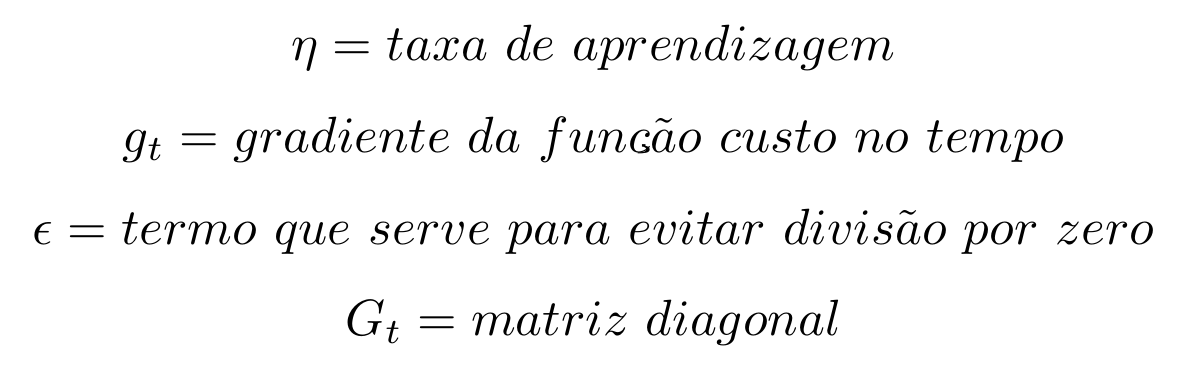
****

Os algoritmos: AdaGrad, RMSProp e Adam são inseridos na categoria dos algoritmos de taxa de aprendizagem adaptativa. Nesta categoria, a taxa de aprendizagem é modificada de uma forma específica em cada um desses algoritmos.

* **AdaGrad**

O Adagrad ajusta a taxa de aprendizagem conforme a frequência dos parâmetros. A taxa de aprendizagem resultante dos parâmetros mais frequentes são atualizadas mais constantemente, contudo com valores menores. No caso dos parâmetros de ocorrência menos frequentes, a taxa de aprendizagem é atualizada com valores maiores. Esta característica torna o algoritmo Adagrad propício a trabalhar com dados esparsos. O vetor de parâmetros é atualizado de acordo com a fórmula:



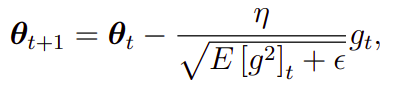


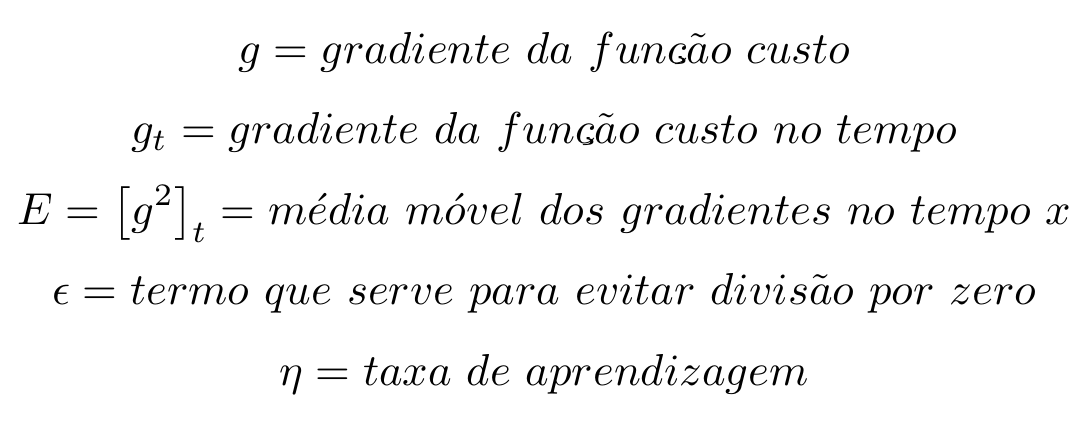
**Vantagem:** Não é necessário ajustar a taxa de aprendizagem manualmente.

**Desvantagem:** O acúmulo dos gradientes pode causar a estagnação da aprendizagem, diminuindo a acurácia para um número elevado de iterações.

* **RMSProp**

Substitui a soma dos gradientes, presente no algoritmo Adagrad, por uma média móvel dos quadrados dos gradientes.

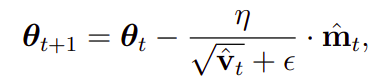
****

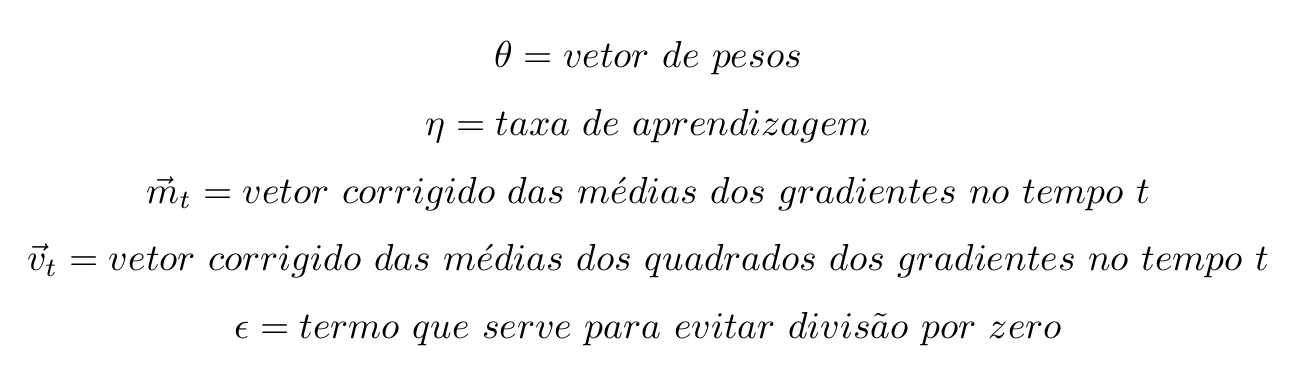
****

**Vantagem:** A média móvel diminui o impacto da soma dos gradientes, evitando o problema da estagnação que ocorre no algoritmo Adagrad.

* **Adam (baseado no método do gradiente estocástico SGM)**

O algoritmo Adam usa as médias móveis dos gradientes e dos quadrados dos gradientes com objetivo de ajustar os parâmetros.





**Vantagem:**

* Converge mais rápido que o algoritmo Adagrad para redes neurais convolucionais
* Possui eficiência igual ou superior aos algoritmos Adagrad e RMSProp