

## Lista de Exercícios 12 – TP555 Inteligência Artificial e Machine Learning

Aluno: Bruno Ferreira Gomes

Matrícula:842

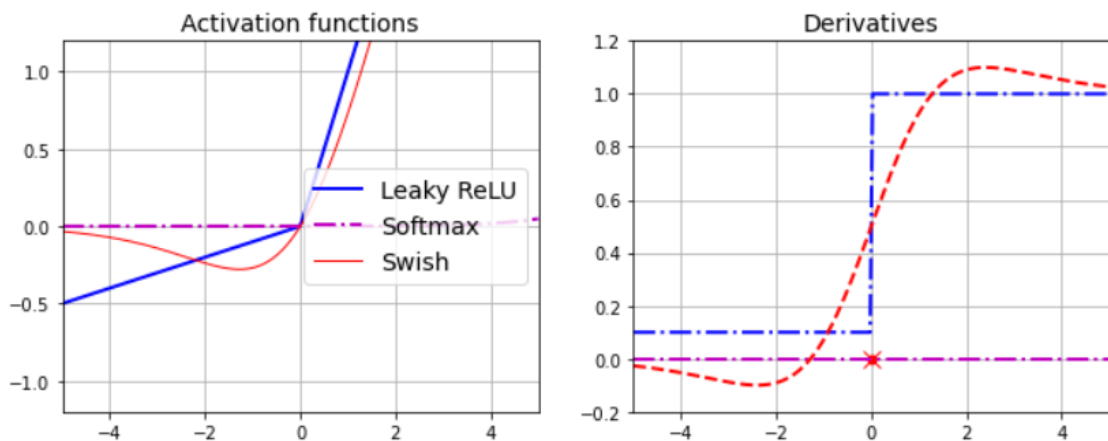
### Ex1)

O principal motivo do uso da função logística sigmoid foi pelo motivo de quando é feita a derivada, o resultado fica sempre diferente de zero, e longe deste. Assim, podemos utilizar esta combinação com o algoritmo do gradiente descendente, permitindo a convergência e combinação do algoritmo com a resposta da função de ativação.

### Ex2)

Pode-se ser citado como dentre as funções de ativação, as seguintes:

- Softmax
- Swish
- Leaky ReLU



Coloquei o código python na pasta Lista12 com esses plots!

### Ex3)

- À partir dos dados informados, devemos dizer que esta matriz de entrada depende do tamanho do batch a ser utilizado na fase de treinamento e do número de neurônios utilizados na camada de entrada, assim, possuindo dimensões TB x 10.
- O vetor de pesos depende do número de neurônios na entrada, e as ligações destes com os neurônios na hidden layer. Assim, podemos dizer que a matriz

de pesos  $W_h$  será de  $10 \times 50$ . Já o vetor de bias  $b_0$ , este é adicionado um por neurônio da hidden layer, sendo assim, este terá dimensões  $50 \times 1$ .

- c) O vetor de pesos da output layer depende dos 50 neurônios da hidden layer e dos 3 neurônios da output layer, sendo assim,  $W_0$  teria dimensões  $50 \times 3$ . Já o vetor bias, este deve ser adicionado um por neurônio, como escrito também para o  $W_h$  no item b), portanto, para o  $b_0$  teremos dimensões  $3 \times 1$ .
- d) A matriz de saída dependerá também do tamanho do batch, e do número de neurônios da output layer, sendo assim, composta das dimensões  $TB \times 3$ .
- e)  $Y = \text{ReLU} ( \text{ReLU}(X*W_h + B_h)*W_o + B_o ) \rightarrow$  Assim como a dada na prova, porém com a função ReLu ao invés da sigmóid.

#### **Ex4)**

Somente com o uso de um neurônio na output layer, já conseguimos implementar esta tarefa de classificação de e-mails. A função de ativação usada pode ser a logística.

Já para solucionar o problema de classificação dos dados MNIST, essa é uma tarefa mais complicada devido o problema ser multiclases, sendo necessários um número maior de neurônios. Podemos dizer que com pelo menos 10 neurônios, e utilizando a função de ativação softmax, tal tarefa pode ser efetuada.

#### **Ex5)**

Os hiperparâmetros que podem ser modificados em uma MPL são: 1) o número de hidden layers, 2) o número de perceptrons/neurônios nestas hidden layers, 3) As funções de ativação e processamento utilizadas tanto na hidden layer como na output layer.

Se a MPL esta sobreajustando, devemos reduzir os dois primeiros parâmetros citados, de modo que consigamos reduzir este sobreajuste e chegar em um modelo que mitiga esse problema e converge na solução mais viável.