

# parcial

April 6, 2022

## 1 Parcial 1

### 1.0.1 Profesor Julio Omar Palacio Niño

### 1.0.2 Estudiante Jesús Ernesto Suárez Triana

### 1.1 1) Preguntas de selección multiple (5 puntos)

De las siguientes sentencias, ¿Cuáles son afirmativas? (puede existir mas de una)

- a) la capacidad de aprendizaje de una red neuronal depende de la aleatoriedad de los pesos
- b) la capacidad de aprendizaje de una red neuronal depende de la variedad y cantidad de ejemplos
- c) las redes neuronales son sistemas que aprenden por medio de los ejemplos
- d) el problema del sobre-aprendizaje se prodouce únicamente el los modelos basados en redes neuronales

#### Respuestas:

- a) False
- b) True
- c) True
- d) False

### 1.2 2) Preguntas de selección multiple (5 puntos)

De las siguientes sentencias, ¿Cuál es afirmativa?

- a) En una red multicapa el ajuste que se realiza en la ultima capa se realiza por medio del delta de la neurona de destino y la activación de la neurona de origen
- b) El ajuste de los pesos de la primera capa de entrada en una red neuronal multicapa se realiza utilizando únicamente el error medido en la salida de toda la red y la activación de la neurona oculta.
- c) En una red multicapa el ajuste de los pesos de la capa de entrada a la primera capa oculta se realiza utilizando el delta de la neurona oculta y todas las activaciones de las neuronas de entrada.

#### Respuestas:

- b) El ajuste de los pesos de la primera capa de entrada en una red neuronal multicapa se realiza utilizando únicamente el error medido en la salida de toda la red y la activación de la neurona oculta.

### 1.3 3) Responda las siguientes preguntas (15 puntos)

- ¿En que consiste el aprendizaje por medio de una red neuronal?.
- ¿Qué es el sobreaprendizaje de una red neuronal?
- ¿Cuál es el significado de los pesos y el sesgo en un perceptrón simple?

Respuestas:

- ¿En que consiste el aprendizaje por medio de una red neuronal?.

El aprendizaje por medio de una red neuronal intenta simular el aprendizaje de una neurona, que basicamente se reduce a si la neurona se excita o no. Para el contexto de aprendizaje profundo, dado un dataset, construimos una red, para que ella se “excite” cuando los datos se parezcan al fenomeno de estudio o no se exciten en caso contrario.

Esto se puede relacionar a los temas de clasificación, regresión, detección, entre otros. donde excitar es activar la neurona a un fenomeno (presencia de un virus, clasificación de grupos, etc).

- ¿Qué es el sobreaprendizaje de una red neuronal?

Es un problema en los procesos de entrenamiento, cuando la red neuronal reproduce muy bien los datos de salida para muestra de Entrenamiento. Esto puede producir que cuando probemos con dataset de test, el error sea grande ya que la red aprendio muy bien solo con el conjunto de entrenamiento.

- ¿Cuál es el significado de los pesos y el sesgo en un perceptrón simple?

Consideremos un fenomeno que esta compuesto para cada individuo  $i$  por características  $x_1, \dots, x_n$  ( $n \geq 1$ ) y una respuesta para cada registro  $y_i$ . Los pesos: Corresponden al valor numérico  $w_j$  ( $1 \leq j \leq n$ ) que permiten dar la relevancia de la característica  $x_j$  para activar (o excitar) la neurona. El sesgo: El sesgo, es ese valor numérico que controla que tan predispuesta esta la neurona a activarse (excitarse) independientemente del valor de los pesos.

### 1.4 4) Responda las siguientes preguntas (15 puntos)

- Indique 3 metricas para evaluar la calidad de una red neuronal cuando resuleve un problema de clasificación
- Indique dos técnicas de preprocesamiento sobre los datos antes de emplearlos en el proceso de entrenamiento de una red neuronal
- ¿Por qué en el proceso de aprendizaje de una red multicapa puede detenerse al encontrar un mínimo local?

Respuestas

- Indique 3 metricas para evaluar la calidad de una red neuronal cuando resuleve un problema de clasificación
  - Precisión  $TP / (TP + FP)$
  - recall  $TP / (TP + FN)$
  - f1-Score  $2(Recall \cdot precision) / (recall + precision)$

- Indique dos técnicas de preprocesamiento sobre los datos antes de emplearlos en el proceso de entrenamiento de una red neuronal.

- Eliminar Duplicados.
- Creación de nuevas Features.
- Codificación de datos categoricos
- División de datos de prueba y de entrenamiento

- ¿Por qué en el proceso de aprendizaje de una red multicapa puede detenerse al encontrar un mínimo local?

Por lo general, cuando entrenamos una red multicapa, esto lo realizamos con procesos de batch\_size y epoch, que permiten trabajar fragmentos de volumen de datos (Puntos en el dominio de estudio) y diferentes iteraciones para calibrar la red.

En el caso de que a diferentes epocas o diferentes batch, no haya variación en las metricas de medición del modelo (Como el accuracy para un modelo de clasificación o RMS para modelos de regresión) podriamos pausar el modelo ya que encontramos un mínimo local, y puede pasar que el dominio de estudio tenga varios minimos locales y no se tengan avance, por la metodologia de optimización que elijamos (COmo gradiente descendiente estocastico).

## 2 Problemas

### 2.1 Problema 1 (30 puntos)

Se desea analizar los siguientes datos por medio de una perceptrón simple

```
[1]: import pandas as pd
```

```
[2]: df = pd.DataFrame({
      'area': [52, 65, 47, 57],
      'conjunto': ['margarita', 'girasol', 'girasol', 'margarita'],
      'valor': [95, 130, 108, 126]
    })
df
```

```
[2]:
```

	area	conjunto	valor
0	52	margarita	95
1	65	girasol	130
2	47	girasol	108
3	57	margarita	126

```
[3]: #categoria de la respuesta, vamos a apuntarle a girasol como cero
df = df.assign(conjunto_num= df.conjunto.apply(lambda x: 1 if x=='margarita'
↪else 0) )
df
```

```
[3]:
```

	area	conjunto	valor	conjunto_num
0	52	margarita	95	1
1	65	girasol	130	0

2	47	girasol	108	0
3	57	margarita	126	1

### 2.1.1 a) escriba la ecuación de salida

$$z = f(0.1\text{area} + 0.4\text{conjunto} + 2*\text{valor} + 1)$$

donde

$f(x) = 1_{\{x>0\}}(x)$  Es la función escalon

### 2.1.2 b) Calcule el valor de salida de la neurona para los valores del problema, considerando tanto la función lineal y la función escalon como activación

```
[4]: def calculo_z ( row):
      b = 1
      w = [0.1, 0.4, 2]

      z = w[0]*row.area + w[1]*row.conjunto_num + w[2]*row.valor
      z = z + b
      #funcion escalon
      if z > 0:
          z = 1
      else:
          z = 0
      return z
```

```
[5]: df.assign(z = df.apply(calculo_z, axis=1))
```

```
[5]:   area  conjunto  valor  conjunto_num  z
0    52  margarita    95             1  1
1    65   girasol   130             0  1
2    47   girasol   108             0  1
3    57  margarita   126             1  1
```

### 2.1.3 c) Que se puede concluir de los valores obtenidos,

#### Respuesta

Dado que el sesgo es 1, y todos los valores dentro de los datos son positivos, podemos observar que el sesgo predispone a la neurona a activarse, independientemente del valor de los pesos.

Si los pesos afectaran negativamente a la variable, podríamos ver un cambio adverso a la función de activación.

## 2.2 2) Problema 2 (30 puntos)

Dado el siguiente problema

Construya un perceptrón simple que permita resolver el problema y resuleva una época completa para ajustar los pesos

**Nota:** una época es el recorrido completo de todos los ejemplos de entrenamientos

```
[6]: df = pd.DataFrame({
      'x1': [0,0,1,1],
      'x2': [0,1,0,1],
      'y': [0,1,1,1],
    })
df
```

```
[6]:   x1  x2  y
0    0   0  0
1    0   1  1
2    1   0  1
3    1   1  1
```

```
[7]: def perceptron_simple(x1,x2,y,b,w1,w2,alpha, n_sample):
      cont=0
      #error = True
      #while error:
      #    error = False
      for i in range(n_sample):
          z = ((x1[i] * w1)+(x2[i] * w2)) + b # calculamos z
          #funcion escalón
          z = z # lineal
          #verificación del error
          if z != y[i]:
              errores = True
              # calcular errores
              error = (y[i] - z)
              # ajustar sesgo
              b = b + ((alpha * error))
              # ajustar pesos
              w1 = w1 + (x1[i] * error * alpha)
              w2 = w2 + (x2[i] * error * alpha)
          # cont=cont+1
      return w1, w2, b, cont
```

```
[8]: w = [3,4]
      b= -7
      alpha = 1
      w1, w2, b, cont = perceptron_simple(x1= df.x1,
                                           x2= df.x2,
                                           y= df.y,
                                           b=b,
                                           w1 = w[0],
```

```
w2 = w[1],  
alpha=alpha,  
n_sample= df.shape[0])
```

```
[9]: print(f'Los pesos obtenidos despues de la primera epoca son: {[w1,w2]}')  
      print(f'El sesgo despues de la primera epoca es de {b}')
```

Los pesos obtenidos despues de la primera epoca son: [2, -1]

El sesgo despues de la primera epoca es de -4